Short proofs of ideal membership

Clemens Hofstadler[1] and Thibaut Verron[2]

CASC 2024
Rennes, France, 5 September 2024

1. Institute for Symbolic Artificial Intelligence, JKU Linz, Austria
2. Institute for Algebra, JKU Linz, Austria

JOHANNES KEPLER UNIVERSITY LINZ

Der Wissenschaftsfonds.

Automated theorem proving

Claim Invertible matrix A with inverse B and inner inverse C

$$\Rightarrow \quad B = C$$

Automated theorem proving

$ACA = A$

Claim Invertible matrix A with inverse B and inner inverse C

$$\Rightarrow \quad B = C$$

Automated theorem proving

$ACA = A$

Claim Invertible matrix A with inverse B and inner inverse C

$$\Rightarrow \quad B = C$$

Proof $B = BAB = BACAB = CAB = C$

Automated theorem proving

$ACA = A$

Claim Invertible matrix A with inverse B and inner inverse C

$$\Rightarrow \quad B = C$$

Proof $B = BAB = BACAB = CAB = C$

Computer algebra approach

Automated theorem proving

$ACA = A$

Claim Invertible matrix A with inverse B and $\boxed{\text{inner inverse } C}$

$$\Rightarrow \quad B = C$$

Proof $$B = BAB = BACAB = CAB = C$$

Computer algebra approach

Matrices $\quad\longrightarrow\quad$ noncommutative indeterminates a, b, c

Automated theorem proving

$ACA = A$

Claim Invertible matrix A with inverse B and $\boxed{\text{inner inverse } C}$

$$\Rightarrow \quad B = C$$

Proof $$B = BAB = BACAB = CAB = C$$

Computer algebra approach

Matrices $\longrightarrow$ noncommutative indeterminates a, b, c

Identities $\longrightarrow$ polynomials in free algebra $K\langle a, b, c\rangle$

Automated theorem proving

$ACA = A$

Claim Invertible matrix A with inverse B and $\boxed{\text{inner inverse } C}$

$$\Rightarrow \quad B = C$$

Proof $$B = BAB = BACAB = CAB = C$$

Computer algebra approach

Matrices	$\longrightarrow$	noncommutative indeterminates a, b, c
Identities	$\longrightarrow$	polynomials in free algebra $K\langle a, b, c\rangle$
Axioms	$\longrightarrow$	ideal $I = \langle ab - 1,\ ba - 1,\ aca - a\rangle$

Automated theorem proving

$ACA = A$

Claim Invertible matrix A with inverse B and $\boxed{\text{inner inverse } C}$

$$\Rightarrow \quad B = C$$

Proof $$B = BAB = BACAB = CAB = C$$

Computer algebra approach

Matrices	$\rightarrow$	noncommutative indeterminates a, b, c
Identities	$\rightarrow$	polynomials in free algebra $K\langle a, b, c\rangle$
Axioms	$\rightarrow$	ideal $I = \langle ab - 1,\ ba - 1,\ aca - a\rangle$
Theorem	$\rightarrow$	ideal membership $b - c \stackrel{?}{\in} I$

Automated theorem proving

$ACA = A$

Claim Invertible matrix A with inverse B and inner inverse C

$$\Rightarrow \quad B = C$$

Proof $B = BAB = BACAB = CAB = C$

Computer algebra approach

Matrices	$\longrightarrow$	noncommutative indeterminates a, b, c
Identities	$\longrightarrow$	polynomials in free algebra $K\langle a, b, c\rangle$
Axioms	$\longrightarrow$	ideal $I = \langle ab - 1,\ ba - 1,\ aca - a\rangle$
Theorem	$\longrightarrow$	ideal membership $b - c \overset{?}{\in} I$
Proof	$\longrightarrow$	$b - c$ reduces to zero modulo Gröbner basis of I

Automated theorem proving

$ACA = A$

Claim Invertible matrix A with inverse B and inner inverse C

$$\Rightarrow \quad B = C$$

Proof $B = BAB = BACAB = CAB = C$

Computer algebra approach

Matrices	$\longrightarrow$	noncommutative indeterminates a, b, c
Identities	$\longrightarrow$	polynomials in free algebra $K\langle a, b, c\rangle$
Axioms	$\longrightarrow$	ideal $I = \langle ab - 1,\ ba - 1,\ aca - a\rangle$
Theorem	$\longrightarrow$	ideal membership $b - c \stackrel{?}{\in} I$
Proof	$\longrightarrow$	see code below

```
sage: from operator_gb import *
sage: R.<a,b,c> = FreeAlgebra(QQ)
sage: I = NCIdeal([a*b-1, b*a-1, a*c*a-a])
sage: I.reduced_form(b-c)
                              0
```

Automated theorem proving

$ACA = A$

Claim Invertible matrix A with inverse B and inner inverse C

$$\Rightarrow \quad B = C$$

Proof $B = BAB = BACAB = CAB = C$

Computer algebra approach

Matrices $\longrightarrow$ noncommutative indeterminates a, b, c

Identities $\longrightarrow$ polynomials in free algebra $K\langle a, b, c\rangle$

Axioms $\longrightarrow$ ideal $I = \langle ab - 1,\ ba - 1,\ aca - a\rangle$

Theorem $\longrightarrow$ ideal membership $b - c \overset{?}{\in} I$

Proof $\longrightarrow$

```
sage: from operator_gb import *
sage: R.<a,b,c> = FreeAlgebra(QQ)
sage: I = NCIdeal([a*b-1, b*a-1, a*c*a-a])
sage: I.reduced_form(b-c)
0
```

Actual proof $\longrightarrow$ cofactor representation

$$b - c = c(ab - 1) + (ba - 1)cab - (ba - 1)b - b(aca - a)b$$

A not so trivial example

Theorem (Djordjević, Dinčić '09) A, B matrices such that AB exists.

$$B^\dagger (ABB^\dagger)^\dagger = (A^\dagger AB)^\dagger A^\dagger = B^\dagger A^\dagger \quad \Rightarrow \quad (AB)^\dagger = B^\dagger A^\dagger$$

A not so trivial example

Theorem (Djordjević, Dinčić '09) A, B matrices such that AB exists.

$$B^\dagger(ABB^\dagger)^\dagger = (A^\dagger AB)^\dagger A^\dagger = B^\dagger A^\dagger \quad \Rightarrow \quad (AB)^\dagger = B^\dagger A^\dagger$$

Correctness of this theorem translates into

$$(ab)^\dagger - b^\dagger a^\dagger \in (f_1, \ldots, f_{44})$$

A not so trivial example

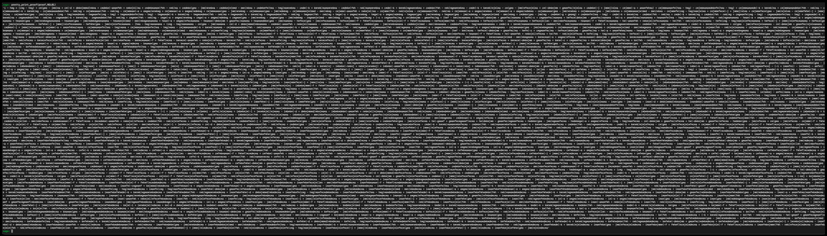
Theorem (Djordjević, Dinčić '09) A, B matrices such that AB exists.

$$B^\dagger (ABB^\dagger)^\dagger = (A^\dagger AB)^\dagger A^\dagger = B^\dagger A^\dagger \quad \Rightarrow \quad (AB)^\dagger = B^\dagger A^\dagger$$

Correctness of this theorem translates into

$$(ab)^\dagger - b^\dagger a^\dagger \in (f_1, \ldots, f_{44})$$

Proof



A not so trivial example

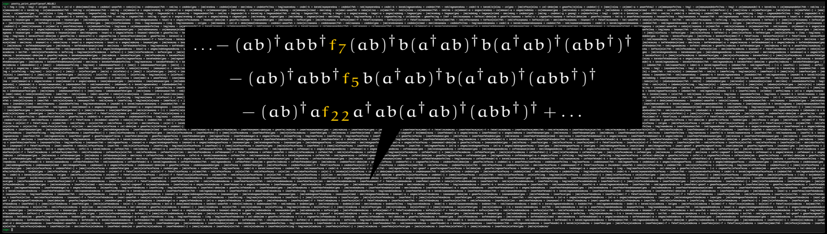
Theorem (Djordjević, Dinčić '09) A, B matrices such that AB exists.

$$B^\dagger(ABB^\dagger)^\dagger = (A^\dagger AB)^\dagger A^\dagger = B^\dagger A^\dagger \quad \Rightarrow \quad (AB)^\dagger = B^\dagger A^\dagger$$

Correctness of this theorem translates into

$$(ab)^\dagger - b^\dagger a^\dagger \in (f_1, \ldots, f_{44})$$

Proof



$$\ldots - (ab)^\dagger abb^\dagger f_7 (ab)^\dagger b(a^\dagger ab)^\dagger b(a^\dagger ab)^\dagger (abb^\dagger)^\dagger$$

$$- (ab)^\dagger abb^\dagger f_5 b(a^\dagger ab)^\dagger b(a^\dagger ab)^\dagger (abb^\dagger)^\dagger$$

$$- (ab)^\dagger a f_{22} a^\dagger ab(a^\dagger ab)^\dagger (abb^\dagger)^\dagger + \ldots$$

A not so trivial example

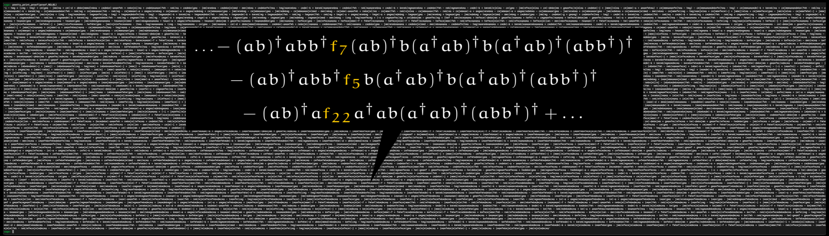
Theorem (Djordjević, Dinčić '09) A, B matrices such that AB exists.

$$B^\dagger(ABB^\dagger)^\dagger = (A^\dagger AB)^\dagger A^\dagger = B^\dagger A^\dagger \quad \Rightarrow \quad (AB)^\dagger = B^\dagger A^\dagger$$

Correctness of this theorem translates into

Proof

$$(ab)^\dagger - b^\dagger a^\dagger \in (f_1, \ldots, f_{44})$$



$$\ldots - (ab)^\dagger abb^\dagger f_7 (ab)^\dagger b(a^\dagger ab)^\dagger b(a^\dagger ab)^\dagger (abb^\dagger)^\dagger$$
$$- (ab)^\dagger abb^\dagger f_5 b(a^\dagger ab)^\dagger b(a^\dagger ab)^\dagger (abb^\dagger)^\dagger$$
$$- (ab)^\dagger a f_{22} a^\dagger ab(a^\dagger ab)^\dagger (abb^\dagger)^\dagger + \ldots$$

Another proof

$$\begin{aligned}
(ab)^\dagger - b^\dagger a^\dagger = {} & f_{21} - f_{10} + b^\dagger f_{14} - f_{12}(ab)^\dagger - b^\dagger(abb^\dagger)^\dagger f_{11} + b^\dagger(abb^\dagger)^\dagger f_{15} \\
& + (a^\dagger ab)^\dagger a^\dagger f_9 (ab)^\dagger - b^* f_{23}((ab)^\dagger)^*(ab)^\dagger - f_{21} ab(ab)^\dagger + f_{22} ab(ab)^\dagger \\
& - f_{39}(a^\dagger)^*((ab)^\dagger)^*(ab)^\dagger + b^\dagger(abb^\dagger)^\dagger((abb^\dagger)^\dagger)^*(b^\dagger)^* f_{31} - b^\dagger f_{14} d^* b^*(a^\dagger)^* \\
& + (a^\dagger ab)^\dagger a^\dagger ab f_{12}(ab)^\dagger - b^\dagger(abb^\dagger)^\dagger f_{15}((ab)^\dagger)^* b^*(a^\dagger)^* \\
& + f_{20} b^*(a^\dagger)^*((ab)^\dagger)^*(ab)^\dagger + (a^\dagger ab)^\dagger a^\dagger abb^* f_{23}((ab)^\dagger)^*(ab)^\dagger
\end{aligned}$$

A not so trivial example

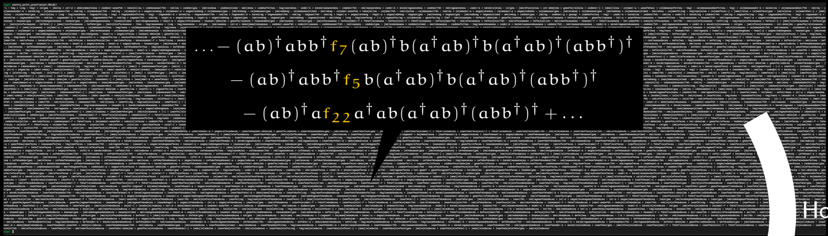
Theorem (Djordjević, Dinčić '09) A, B matrices such that AB exists.

$$B^\dagger(ABB^\dagger)^\dagger = (A^\dagger AB)^\dagger A^\dagger = B^\dagger A^\dagger \quad \Rightarrow \quad (AB)^\dagger = B^\dagger A^\dagger$$

Correctness of this theorem translates into

$$(ab)^\dagger - b^\dagger a^\dagger \in (f_1, \ldots, f_{44})$$

Proof



How?

Another proof

$$\begin{aligned}(ab)^\dagger - b^\dagger a^\dagger = {} & f_{21} - f_{10} + b^\dagger f_{14} - f_{12}(ab)^\dagger - b^\dagger (abb^\dagger)^\dagger f_{11} + b^\dagger (abb^\dagger)^\dagger f_{15} \\ & + (a^\dagger ab)^\dagger a^\dagger f_9 (ab)^\dagger - b^* f_{23}((ab)^\dagger)^*(ab)^\dagger - f_{21} ab(ab)^\dagger + f_{22} ab(ab)^\dagger \\ & - f_{39}(a^\dagger)^*((ab)^\dagger)^*(ab)^\dagger + b^\dagger (abb^\dagger)^\dagger ((abb^\dagger)^\dagger)^*(b^\dagger)^* f_{31} - b^\dagger f_{14} d^* b^* (a^\dagger)^* \\ & + (a^\dagger ab)^\dagger a^\dagger ab f_{12}(ab)^\dagger - b^\dagger (abb^\dagger)^\dagger f_{15}((ab)^\dagger)^* b^* (a^\dagger)^* \\ & + f_{20} b^* (a^\dagger)^* ((ab)^\dagger)^* (ab)^\dagger + (a^\dagger ab)^\dagger a^\dagger abb^* f_{23}((ab)^\dagger)^*(ab)^\dagger\end{aligned}$$

Problem formulation

⚠ All results of this talk also apply to the commutative case!

Problem formulation

⚠ All results of this talk also apply to the commutative case!

Problem

Given $f, f_1, \ldots, f_r \in K\langle X\rangle$, $N \in \mathbb{N}$

Compute $a_i, b_i \in \langle X\rangle$, $c_i \in K$ such that

$$f = \sum_{i=1}^{\leqslant N} c_i a_i \cdot f_{j_i} \cdot b_i,$$

if existent, else return $\textsc{Failed}$.

Problem formulation

⚠ All results of this talk also apply to the commutative case!

Problem

Given $f, f_1, \ldots, f_r \in K\langle X\rangle$, $N \in \mathbb{N}$

Compute $a_i, b_i \in \langle X\rangle$, $c_i \in K$ such that

$$f = \sum_{i=1}^{\leqslant N} c_i a_i \cdot f_{j_i} \cdot b_i,$$

if existent, else return $\textsc{Failed}$.

Question Is this problem decidable?

Problem formulation

⚠ All results of this talk also apply to the commutative case!

Problem

Given $f, f_1, \ldots, f_r \in K\langle X\rangle$, $N \in \mathbb{N}$

Compute $a_i, b_i \in \langle X\rangle$, $c_i \in K$ such that

$$f = \sum_{i=1}^{\leqslant N} c_i a_i \cdot f_{j_i} \cdot b_i,$$

if existent, else return $\textsc{Failed}$.

Question Is this problem decidable?

Fact: Ideal membership for noncommutative polynomials is undecidable

Thus, decidability is, a priori, not clear.

Problem formulation

⚠ All results of this talk also apply to the commutative case!

Problem

Given $f, f_1, \dots, f_r \in K\langle X\rangle$, $N \in \mathbb{N}$

Compute $a_i, b_i \in \langle X\rangle$, $c_i \in K$ such that

$$f = \sum_{i=1}^{\leqslant N} c_i a_i \cdot f_{j_i} \cdot b_i,$$

if existent, else return FAILED.

Question Is this problem decidable?

Fact: Ideal membership for noncommutative polynomials is undecidable

Thus, decidability is, a priori, not clear.

Difficulty: no degree bound for a_i and b_i

Problem formulation

All results of this talk also apply to the commutative case!

Problem

Given $f, f_1, \ldots, f_r \in K\langle X \rangle$, $N \in \mathbb{N}$

Compute $a_i, b_i \in \langle X \rangle$, $c_i \in K$ such that

$$f = \sum_{i=1}^{\leqslant N} c_i a_i \cdot f_{j_i} \cdot b_i,$$

if existent, else return $\textsc{Failed}$.

Question Is this problem decidable?

Fact: Ideal membership for noncommutative polynomials is undecidable

Thus, decidability is, a priori, not clear.

Difficulty: no degree bound for a_i and b_i

Theorem (H., Verron '24) The problem is decidable!

Polynomial rewriting

We all know polynomial reduction $f \to_g f'$, where we use the leading term of g to cancel a term in f.

Polynomial rewriting

We all know polynomial ~~reduction~~ rewriting $f \to_g f'$, where we use ~~the leading~~ a term term of g to ~~cancel~~ modify a term in f.

Polynomial rewriting

We all know polynomial ~~reduction~~ rewriting $f \to_g f'$, where we use ~~the leading term~~ a term of g to ~~cancel~~ modify a term in f.

Example: Consider $f = xy^2 + z$ and $g = y^2 + z$.

- $f - xg = z - xz$ is a reduction (and thus also a rewriting)

Polynomial rewriting

We all know polynomial ~~reduction~~ rewriting $f \to_g f'$, where we use ~~the leading~~ a term of g to ~~cancel~~ modify a term in f.

Example: Consider $f = xy^2 + z$ and $g = y^2 + z$.

- $f - xg = z - xz$ is a reduction (and thus also a rewriting)
- $f + g = xy^2 + 2z + y^2$ is a rewriting

Polynomial rewriting

We all know polynomial ~~reduction~~ rewriting $f \to_g f'$, where we use ~~the leading term~~ a term of g to ~~cancel~~ modify a term in f.

Example: Consider $f = xy^2 + z$ and $g = y^2 + z$.

- $f - xg \;=\; z - xz$ is a reduction (and thus also a rewriting)
- $f + g \;=\; xy^2 + 2z + y^2$ is a rewriting
- $f + yg \;=\; xy^2 + z + y^3 + yz$ is **not** a rewriting

Polynomial rewriting

We all know polynomial ~~reduction~~ rewriting $f \to_g f'$, where we use ~~the leading term of~~ a term of g to ~~cancel~~ modify a term in f.

Example: Consider $f = xy^2 + z$ and $g = y^2 + z$.

- $f - xg = z - xz$ is a reduction (and thus also a rewriting)
- $f + g = xy^2 + 2z + y^2$ is a rewriting
- $f + yg = xy^2 + z + y^3 + yz$ is **not** a rewriting

Facts

- $f \in \langle f_1, \ldots, f_r \rangle$ iff f can be rewritten to 0 by $\{f_1, \ldots, f_r\}$
- f has minimal representation with N terms iff f can be rewritten to 0 in N steps
- We can bound the degree of terms appearing in a rewriting step!

A first algorithm

Theorem (H., Verron '24) Let $f, f_1, \ldots, f_r \in K\langle X\rangle$ and $N \in \mathbb{N}$. If there exists a minimal representation

$$f = \sum_{i=1}^{\leqslant N} c_i a_i \cdot f_{j_i} \cdot b_i,$$

then $\deg(a_i f_{j_i} b_i) \leqslant D \coloneqq \mathrm{poly}(f, f_1, \ldots, f_r, N)$.

A first algorithm

Theorem (H., Verron '24) Let $f, f_1, \ldots, f_r \in K\langle X\rangle$ and $N \in \mathbb{N}$. If there exists a minimal representation

$$f = \sum_{i=1}^{\leqslant N} c_i a_i \cdot f_{j_i} \cdot b_i,$$

then $\deg(a_i f_{j_i} b_i) \leqslant D \coloneqq \mathrm{poly}(f, f_1, \ldots, f_r, N)$.

A first algorithm

1. Make ansatz
$$f = \sum_i c_i a_i \cdot f_{j_i} \cdot b_i$$
with unknown $c_i \in K$ and all $a_i, b_i \in \langle X\rangle$ with $\deg(a_i f_{j_i} b_i) \leqslant D$.
2. Look for solution of the resulting linear system with $\leqslant N$ nonzero coordinates.

A first algorithm

Theorem (H., Verron '24) Let $f, f_1, \ldots, f_r \in K\langle X\rangle$ and $N \in \mathbb{N}$. If there exists a minimal representation

$$f = \sum_{i=1}^{\leqslant N} c_i a_i \cdot f_{j_i} \cdot b_i,$$

then $\deg(a_i f_{j_i} b_i) \leqslant D \coloneqq \text{poly}(f, f_1, \ldots, f_r, N)$.

A first algorithm

1. Make ansatz
$$f = \sum_{i} c_i a_i \cdot f_{j_i} \cdot b_i$$
with unknown $c_i \in K$ and all $a_i, b_i \in \langle X\rangle$ with $\deg(a_i f_{j_i} b_i) \leqslant D$.
2. Look for solution of the resulting linear system with $\leqslant N$ nonzero coordinates.

Observations

- Step 1 yields huge but finite system
- Step 2 is difficult but decidable
- The algorithm is not practical for non-trivial examples

A first algorithm

Theorem (H., Verron '24) Let $f, f_1, \ldots, f_r \in K\langle X \rangle$ and $N \in \mathbb{N}$. If there exists a minimal representation

$$f = \sum_{i=1}^{\leqslant N} c_i a_i \cdot f_{j_i} \cdot b_i,$$

then $\deg(a_i f_{j_i} b_i) \leqslant D \coloneqq \mathrm{poly}(f, f_1, \ldots, f_r, N)$.

A first algorithm

1. Make ansatz

$$f = \sum_i c_i a_i \cdot f_{j_i} \cdot b_i$$

with unknown $c_i \in K$ and all $a_i, b_i \in \langle X \rangle$ with $\deg(a_i f_{j_i} b_i) \leqslant D$.

2. Look for solution of the resulting linear system with $\leqslant N$ nonzero coordinates.

Observations

- Step 1 yields huge but finite system $\rightsquigarrow$ signatures to reduce search space
- Step 2 is difficult but decidable
- The algorithm is not practical for non-trivial examples

A first algorithm

Theorem (H., Verron '24) Let $f, f_1, \ldots, f_r \in K\langle X\rangle$ and $N \in \mathbb{N}$. If there exists a minimal representation

$$f = \sum_{i=1}^{\leqslant N} c_i a_i \cdot f_{j_i} \cdot b_i,$$

then $\deg(a_i f_{j_i} b_i) \leqslant D \coloneqq \mathrm{poly}(f, f_1, \ldots, f_r, N)$.

A first algorithm

1. Make ansatz
$$f = \sum_i c_i a_i \cdot f_{j_i} \cdot b_i$$
with unknown $c_i \in K$ and all $a_i, b_i \in \langle X\rangle$ with $\deg(a_i f_{j_i} b_i) \leqslant D$.
2. Look for solution of the resulting linear system with $\leqslant N$ nonzero coordinates.

Observations

- Step 1 yields huge but finite system $\rightsquigarrow$ signatures to reduce search space
- Step 2 is difficult but decidable $\rightsquigarrow$ linear programming to approx. solution
- The algorithm is not practical for non-trivial examples

Sparse solutions of a linear system

Min-RVLS (Minimum Relevant Variables in Linear Systems)

Given	$A \in \mathbb{Q}^{m \times n}, \mathbf{b} \in \mathbb{Q}^m, N \in \{0, \ldots, n\}$
Compute	solution $\mathbf{y}$ of $A\mathbf{y} = \mathbf{b}$ with $\leqslant N$ nonzero coordinates, if existent

Sparse solutions of a linear system

Min-RVLS (Minimum Relevant Variables in Linear Systems)

Given $A \in \mathbb{Q}^{m \times n}, \mathbf{b} \in \mathbb{Q}^m, N \in \{0, \ldots, n\}$

Compute solution $\mathbf{y}$ of $A\mathbf{y} = \mathbf{b}$ with $\leqslant N$ nonzero coordinates, if existent

Important problem (in signal processing, coding theory, machine learning,...)

Difficult problem (NP-complete)

Sparse solutions of a linear system

Min-RVLS (Minimum Relevant Variables in Linear Systems)

Given $A \in \mathbb{Q}^{m \times n}, \mathbf{b} \in \mathbb{Q}^m, N \in \{0, \ldots, n\}$

Compute solution $\mathbf{y}$ of $A\mathbf{y} = \mathbf{b}$ with $\leqslant N$ nonzero coordinates, if existent

Important problem (in signal processing, coding theory, machine learning,...)

Difficult problem (NP-complete)

Min-RVLS can be reduced to our short representations problem

$\rightsquigarrow$ our problem is also NP-complete

Sparse solutions of a linear system

Min-RVLS (Minimum Relevant Variables in Linear Systems)

Given $A \in \mathbb{Q}^{m \times n}, \mathbf{b} \in \mathbb{Q}^m, N \in \{0, \ldots, n\}$

Compute solution $\mathbf{y}$ of $A\mathbf{y} = \mathbf{b}$ with $\leqslant N$ nonzero coordinates, if existent

Important problem (in signal processing, coding theory, machine learning,...)

Difficult problem (NP-complete)

Min-RVLS can be reduced to our short representations problem

$\rightsquigarrow$ our problem is also NP-complete

Basis pursuit (Chen, Donoho, Saunders '01) $=$ ℓ_1-relaxation

Given $A \in \mathbb{Q}^{m \times n}$, $\mathbf{b} \in \mathbb{Q}^m$

Compute solution $\mathbf{y}$ of $A\mathbf{y} = \mathbf{b}$ with $\|\mathbf{y}\|_1 = \sum_i |\mathbf{y}_i|$ minimal, if existent

Sparse solutions of a linear system

Min-RVLS (Minimum Relevant Variables in Linear Systems)

Given $A \in \mathbb{Q}^{m \times n}, \mathbf{b} \in \mathbb{Q}^m, N \in \{0, \dots, n\}$

Compute solution $\mathbf{y}$ of $A\mathbf{y} = \mathbf{b}$ with $\leqslant N$ nonzero coordinates, if existent

Important problem (in signal processing, coding theory, machine learning,. . .)

Difficult problem (NP-complete)

Min-RVLS can be reduced to our short representations problem

$\rightsquigarrow$ our problem is also NP-complete

Basis pursuit (Chen, Donoho, Saunders '01) $=$ ℓ_1-relaxation

Given $A \in \mathbb{Q}^{m \times n}$, $\mathbf{b} \in \mathbb{Q}^m$

Compute solution $\mathbf{y}$ of $A\mathbf{y} = \mathbf{b}$ with $\|\mathbf{y}\|_1 = \sum_i |\mathbf{y}_i|$ minimal, if existent

Fact: such an ℓ_1-minimal solution can be computed with linear programming

Reducing the search space

So far, search space is

$$B = \left\{ a f_i b \mid a, b \in \langle X \rangle, \deg(a f_i b) \leqslant D \right\}.$$

Reducing the search space

So far, search space is

$$B = \left\{ af_ib \mid a, b \in \langle X \rangle, \deg(af_ib) \leqslant D \right\}.$$

Observation We only need those af_ib that could appear in a rewriting sequence.

Reducing the search space

So far, search space is

$$B = \left\{ a f_i b \mid a, b \in \langle X \rangle, \deg(a f_i b) \leqslant D \right\}.$$

Observation We only need those $a f_i b$ that could appear in a rewriting sequence.

Finding those elements is a combinatorial problem and can be done effectively (symbolic preprocessing (Faugère '99)).

Reducing the search space

So far, search space is

$$B = \{af_ib \mid a, b \in \langle X \rangle, \deg(af_ib) \leqslant D, af_ib \text{ a rewriter}\}.$$

Observation We only need those af_ib that could appear in a rewriting sequence.

Finding those elements is a combinatorial problem and can be done effectively (symbolic preprocessing (Faugère '99)).

Reducing the search space

So far, search space is

$$B = \left\{ af_ib \mid a, b \in \langle X \rangle, \deg(af_ib) \leqslant D, af_ib \text{ a rewriter} \right\}.$$

Observation Rewriting can be extended from polynomials to representations (formally, to bimodule elements).

Reducing the search space

So far, search space is

$$B = \{ a f_i b \mid a, b \in \langle X \rangle, \deg(a f_i b) \leqslant D, a f_i b \text{ a rewriter} \}.$$

Observation Rewriting can be extended from polynomials to representations (formally, to bimodule elements).

Example: Consider $\alpha = xyf_2 + xf_1 + f_2y$ and $\gamma = f_1x - yf_2 + f_1$.

$$\alpha \rightsquigarrow_{\gamma} xf_1x + f_2y$$

Reducing the search space

So far, search space is

$$B = \left\{ af_i b \mid a, b \in \langle X \rangle, \deg(af_i b) \leqslant D, af_i b \text{ a rewriter} \right\}.$$

Observation Rewriting can be extended from polynomials to representations (formally, to bimodule elements).

Example: Consider $\alpha = xyf_2 + xf_1 + f_2 y$ and $\gamma = f_1 x - yf_2 + f_1$.

$$\alpha \rightsquigarrow_{\gamma} xf_1 x + f_2 y$$

Reducing the search space

So far, search space is

$$B = \{af_ib \mid a, b \in \langle X \rangle, \deg(af_ib) \leqslant D, af_ib \text{ a rewriter}\}.$$

Observation Rewriting can be extended from polynomials to representations (formally, to bimodule elements).

Example: Consider $\alpha = xyf_2 + xf_1 + f_2y$ and $\gamma = f_1x - yf_2 + f_1$.

$$\alpha \rightsquigarrow_\gamma xf_1x + f_2y$$

Theorem

(H., Verron '24)

$$\alpha \quad \rightsquigarrow_H \quad \beta$$

Reducing the search space

So far, search space is

$$B = \{af_ib \mid a, b \in \langle X\rangle, \deg(af_ib) \leqslant D, af_ib \text{ a rewriter}\}.$$

Observation Rewriting can be extended from polynomials to representations (formally, to bimodule elements).

Example: Consider $\alpha = xyf_2 + xf_1 + f_2y$ and $\gamma = f_1x - yf_2 + f_1$.

$$\alpha \rightsquigarrow_\gamma xf_1x + f_2y$$

Theorem

(H., Verron '24)

representation of f
with deg $\leqslant$ D

$$\boxed{\alpha} \rightsquigarrow_H \beta$$

Reducing the search space

So far, search space is

$$B = \{ a f_i b \mid a, b \in \langle X \rangle, \deg(a f_i b) \leqslant D, a f_i b \text{ a rewriter} \}.$$

Observation Rewriting can be extended from polynomials to representations (formally, to bimodule elements).

Example: Consider $\alpha = xyf_2 + xf_1 + f_2 y$ and $\gamma = f_1 x - yf_2 + f_1$.

$$\alpha \rightsquigarrow_\gamma xf_1x + f_2y$$

Theorem

(H., Verron '24)

representation of f with deg $\leqslant$ D

minimal repr. of f with deg $\leqslant$ D

$$\boxed{\alpha} \rightsquigarrow_H \boxed{\beta}$$

Reducing the search space

So far, search space is

$$B = \{af_ib \mid a, b \in \langle X \rangle, \deg(af_ib) \leqslant D, af_ib \text{ a rewriter}\}.$$

Observation Rewriting can be extended from polynomials to representations (formally, to bimodule elements).

Example: Consider $\alpha = xyf_2 + xf_1 + f_2y$ and $\gamma = f_1x - yf_2 + f_1$.

$$\alpha \rightsquigarrow_{\gamma} xf_1x + f_2y$$

Theorem

(H., Verron '24)

representation of f with deg $\leqslant$ D

minimal repr. of f with deg $\leqslant$ D

$$\boxed{\alpha} \rightsquigarrow_{\boxed{H}} \boxed{\beta}$$

GB of $\mathrm{Syz}(f_1, \ldots, f_r)$

Reducing the search space

So far, search space is

$$B_\alpha = \{ af_ib \mid a, b \in \langle X \rangle, \deg(af_ib) \leqslant D, af_ib \text{ appears in rewriter of } \alpha \}.$$

Observation Rewriting can be extended from polynomials to representations (formally, to bimodule elements).

Example: Consider $\alpha = xyf_2 + xf_1 + f_2y$ and $\gamma = f_1x - yf_2 + f_1$.

$$\alpha \rightsquigarrow_\gamma xf_1x + f_2y$$

Theorem

(H., Verron '24)

representation of f with deg $\leqslant$ D

minimal repr. of f with deg $\leqslant$ D

$$\boxed{\alpha} \rightsquigarrow_{\boxed{H}} \boxed{\beta}$$

GB of $\mathrm{Syz}(f_1, \ldots, f_r)$

Reducing the search space

So far, search space is

$$B_\alpha = \{ a f_i b \mid a, b \in \langle X \rangle, \deg(a f_i b) \leqslant D, a f_i b \text{ appears in rewriter of } \alpha \}.$$

Observation Rewriting can be extended from polynomials to representations (formally, to bimodule elements).

Example: Consider $\alpha = xyf_2 + xf_1 + f_2y$ and $\gamma = f_1x - yf_2 + f_1$.

$$\alpha \rightsquigarrow_\gamma xf_1x + f_2y$$

Theorem
(H., Verron '24)

representation of f
with deg $\leqslant$ D

minimal repr. of f
with deg $\leqslant$ D

$$\alpha \rightsquigarrow_H \beta$$

GB of $\mathrm{Syz}(f_1, \dots, f_r)$

Problem: $\mathrm{Syz}(f_1, \dots, f_r)$ usually has no finite Gröbner basis

Reducing the search space

So far, search space is

$$B_\alpha = \left\{ a f_i b \mid a, b \in \langle X \rangle, \deg(a f_i b) \leqslant D, \text{ } a f_i b \text{ appears in rewriter of } \alpha \right\}.$$

Observation Rewriting can be extended from polynomials to representations (formally, to bimodule elements).

Example: Consider $\alpha = xyf_2 + xf_1 + f_2y$ and $\gamma = f_1x - yf_2 + f_1$.

$$\alpha \rightsquigarrow_\gamma xf_1x + f_2y$$

Theorem

(H., Verron '24)

$$\underbrace{\alpha}_{\substack{\text{representation of } f\\ \text{with deg} \leqslant D}} \rightsquigarrow_{\underbrace{H}_{\substack{\text{GB of } \mathrm{Syz}(f_1,\dots,f_r)\\ \text{up to deg } D}}} \underbrace{\beta}_{\substack{\text{minimal repr. of } f\\ \text{with deg} \leqslant D}}$$

Problem: $\mathrm{Syz}(f_1, \dots, f_r)$ usually has no finite Gröbner basis

Solution: . . . but Gröbner basis up to degree D is finite

and can be computed using signature-based algorithms (H., Verron '22,'23)

A practical algorithm

Given $f, f_1, \dots, f_r \in \mathbb{Q}\langle X \rangle$, a representation α of f with degree $\leqslant D \in \mathbb{N}$

Compute an ℓ_1-minimal representation of f with degree $\leqslant D$

A practical algorithm

Given $f, f_1, \ldots, f_r \in \mathbb{Q}\langle X \rangle$, a representation α of f with degree $\leqslant D \in \mathbb{N}$

Compute an ℓ_1-minimal representation of f with degree $\leqslant D$

1. Compute Gröbner basis of $\mathrm{Syz}(f_1, \ldots, f_r)$ up to degree D
2. Compute search space $B_\alpha = \{a_1 f_{i_1} b_1, \ldots, a_k f_{i_k} b_k\}$ (sym. preprocessing)
3. Make ansatz for f using B_α and α
4. Compute ℓ_1-minimal solution of the resulting system (linear programming)

A practical algorithm

Given $f, f_1, \ldots, f_r \in \mathbb{Q}\langle X \rangle$, a representation α of f with degree $\leqslant D \in \mathbb{N}$

Compute an ℓ_1-minimal representation of f with degree $\leqslant D$

1. Compute Gröbner basis of $\mathrm{Syz}(f_1, \ldots, f_r)$ up to degree D
2. Compute search space $B_\alpha = \{a_1 f_{i_1} b_1, \ldots, a_k f_{i_k} b_k\}$ (sym. preprocessing)
3. Make ansatz for f using B_α and α
4. Compute ℓ_1-minimal solution of the resulting system (linear programming)

Observations

- Still exponential worst-case complexity, but good behaviour in practice

A practical algorithm

Given $f, f_1, \ldots, f_r \in \mathbb{Q}\langle X\rangle$, a representation α of f with degree $\leqslant D \in \mathbb{N}$

Compute an ℓ_1-minimal representation of f with degree $\leqslant D$

1. Compute Gröbner basis of $\mathrm{Syz}(f_1, \ldots, f_r)$ up to degree D
2. Compute search space $B_\alpha = \{a_1 f_{i_1} b_1, \ldots, a_k f_{i_k} b_k\}$ (sym. preprocessing)
3. Make ansatz for f using B_α and α
4. Compute ℓ_1-minimal solution of the resulting system (linear programming)

Observations

- Still exponential worst-case complexity, but good behaviour in practice
- In general, no guarantee for shortest repr., but good behaviour in practice
- Many examples are even totally unimodular $\rightsquigarrow$ algorithm is guaranteed to return a shortest representation

A practical algorithm

Given $f, f_1, \ldots, f_r \in \mathbb{Q}\langle X\rangle$, a representation α of f with degree $\leqslant D \in \mathbb{N}$

Compute an ℓ_1-minimal representation of f with degree $\leqslant D$

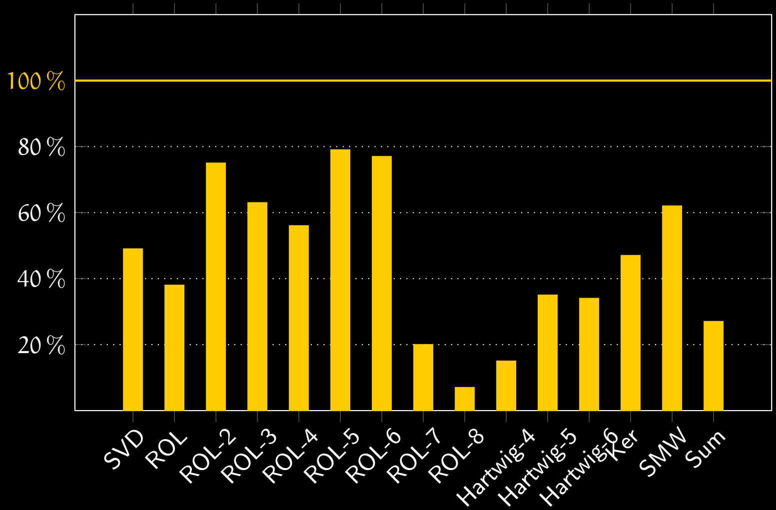
1. Compute Gröbner basis of $\mathrm{Syz}(f_1, \ldots, f_r)$ up to degree D
2. Compute search space $B_\alpha = \{a_1 f_{i_1} b_1, \ldots, a_k f_{i_k} b_k\}$ (sym. preprocessing)
3. Make ansatz for f using B_α and α
4. Compute ℓ_1-minimal solution of the resulting system (linear programming)

Observations

- Still exponential worst-case complexity, but good behaviour in practice
- In general, no guarantee for shortest repr., but good behaviour in practice
- Many examples are even totally unimodular $\rightsquigarrow$ algorithm is guaranteed to return a shortest representation
- The LP approach also opens way to thinner metrics (e.g., ignore certain f_i, weigh terms by degree,...)

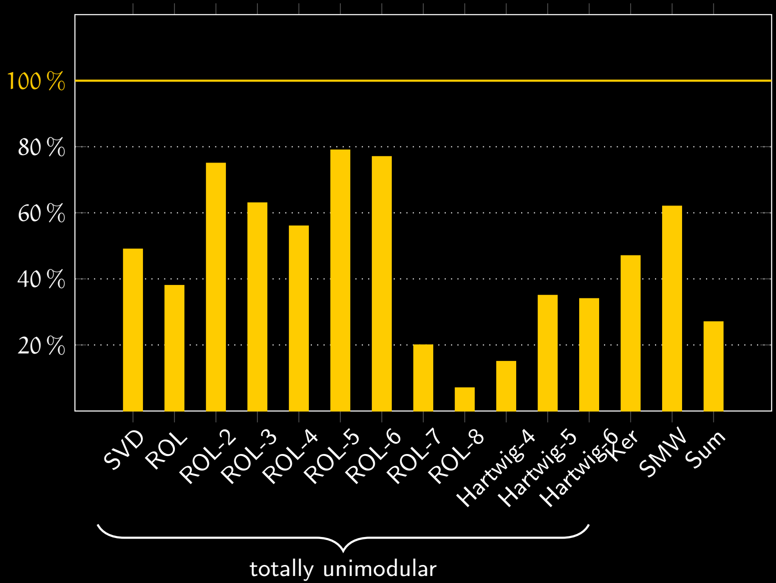
Experiments

Relative improvement on different operator statements with new algorithm



Experiments

Relative improvement on different operator statements with new algorithm



Conclusion

Summary

- Computing shortest proofs of ideal membership is decidable but hard
- Practical algorithm for computing short proofs
- Combination of signature techniques and linear programming
- Also works in the commutative case (!)

Outlook

- More efficient representations using additional generators ("lemmas")
- Analyse behaviour/complexity for particular classes of ideals

Reference

C. Hofstadler and T. Verron. *Short proofs of ideal membership*. Journal of Symbolic Computation 125: 102325, 2024.