Noncommutative Gröbner bases and automated proofs of operator statements

Clemens Hofstadler[1,2]

supervised by Georg Regensburger and Clemens G. Raab

1. Institute for Algebra, Johannes Kepler University Linz, Austria
2. Institute of Mathematics, University of Kassel, Germany

Automated proofs of operator statements

Automated proofs of operator statements

DISCRETE MATHEMATICS AND ITS APPLICATIONS
Series Editor KENNETH H. ROSEN

HANDBOOK OF LINEAR ALGEBRA

SECOND EDITION

$$\begin{bmatrix} 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 6 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 2 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}$$

Edited by
Leslie Hogben

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

5.7 Pseudo-Inverse

Definitions:

A **Moore–Penrose pseudo-inverse** of a matrix $A \in \mathbb{C}^{m\times n}$ is a matrix $A^\dagger \in \mathbb{C}^{n\times m}$ that satisfies the following four **Penrose** conditions:

$$AA^\dagger A = A; \quad A^\dagger AA^\dagger = A^\dagger; \quad (AA^\dagger)^* = AA^\dagger; \quad (A^\dagger A)^* = A^\dagger A.$$

Facts:

All the following facts except those with a specific reference can be found in [Gra83, pp. 105–141] or [RM71, pp. 44–67].

1. Every $A \in \mathbb{C}^{m\times n}$ has a unique pseudo-inverse $A^\dagger$.
2. If $A \in \mathbb{R}^{m\times n}$, then $A^\dagger$ is real.
3. If $A \in \mathbb{C}^{m\times n}$ of rank r has a full rank decomposition $A = BC$, where $B \in \mathbb{C}^{m\times r}$ and $C \in \mathbb{C}^{r\times n}$, then $A^\dagger$ can be evaluated using $A^\dagger = C^*(B^*AC^*)^{-1}B^*$.
4. [LH95, p. 38] If $A \in \mathbb{C}^{m\times n}$ of rank $r \le \min\{m,n\}$ has an SVD $A = U\Sigma V^*$, then its pseudo-inverse is $A^\dagger = V\Sigma^\dagger U^*$, where
$$\Sigma^\dagger = \operatorname{diag}(1/\sigma_1, \ldots, 1/\sigma_r, 0, \ldots, 0) \in \mathbb{R}^{n\times m}.$$
5. [Hig96, p. 412] The pseudo-inverse $A^\dagger$ of $A \in F^{m\times n}$ ($F = \mathbb{C}$ or $\mathbb{R}$) solves the minimization problem
$$\min_{X\in F^{n\times m}} \|AX - I_m\|_F^2.$$
6. $\mathbf{0}_{mn}^\dagger = \mathbf{0}_{nm}$ and $J_{mn}^\dagger = \frac{1}{mn}J_{nm}$, where $\mathbf{0}_{mn} \in \mathbb{C}^{m\times n}$ is the all 0s matrix and $J_{mn} \in \mathbb{C}^{m\times n}$ is the all 1s matrix.
7. If $\mathbf{x} \neq \mathbf{0}$, $\mathbf{y} \neq \mathbf{0}$, then $(\mathbf{x}\mathbf{y}^*)^\dagger = \dfrac{\mathbf{y}\mathbf{x}^*}{\|\mathbf{x}\|^2\|\mathbf{y}\|^2}$.
8. If $\mathbf{x} \neq \mathbf{0}$, then $\mathbf{x}^\dagger = \dfrac{\mathbf{x}^*}{\|\mathbf{x}\|^2}$.
9. Let α be a scalar. Denote
$$\alpha^\dagger = \begin{cases} \alpha^{-1}, & \text{if } \alpha \neq 0, \\ 0, & \text{if } \alpha = 0. \end{cases}$$
Then
   (a) $(\alpha A)^\dagger = \alpha^\dagger A^\dagger$.
   (b) $(\operatorname{diag}(\beta_1, \beta_2, \cdots, \beta_n))^\dagger = \operatorname{diag}(\beta_1^\dagger, \beta_2^\dagger, \cdots, \beta_n^\dagger)$.
10. $(A^\dagger)^* = (A^*)^\dagger$; $(A^\dagger)^\dagger = A$.
11. If A is a nonsingular square matrix, then $A^\dagger = A^{-1}$.
12. If U has orthonormal columns or orthonormal rows, then $U^\dagger = U^*$.
13. If $A = A^*$ and $A = A^2$, then $A^\dagger = A$.
14. $A^\dagger = A^*$ if and only if A^*A is idempotent.
15. If A is normal and k is a positive integer, then $AA^\dagger = A^\dagger A$ and $(A^k)^\dagger = (A^\dagger)^k$.
16. If $U \in \mathbb{C}^{m\times n}$ is of rank n and satisfies $U^\dagger = U^*$, then U has orthonormal columns.
17. If $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are unitary matrices, then $(UAV)^\dagger = V^*A^\dagger U^*$.
18. $A^\dagger = (A^*A)^\dagger A^* = A^*(AA^*)^\dagger$. In particular,
   (a) if $A \in \mathbb{C}^{m\times n}$ $(m \ge n)$ has full rank n, then $A^\dagger = (A^*A)^{-1}A^*$;
   (b) if $A \in \mathbb{C}^{m\times n}$ $(m \le n)$ has full rank m, then $A^\dagger = A^*(AA^*)^{-1}$.
19. Let $A \in \mathbb{C}^{m\times n}$. Then

   (a) $A^\dagger A$, $AA^\dagger$, $I_n - A^\dagger A$, and $I_m - AA^\dagger$ are orthogonal projections.
   (b) $\operatorname{rank}(A) = \operatorname{rank}(A^\dagger) = \operatorname{rank}(AA^\dagger) = \operatorname{rank}(A^\dagger A)$.
   (c) $\operatorname{rank}(I_n - A^\dagger A) = n - \operatorname{rank}(A)$.
   (d) $\operatorname{rank}(I_m - AA^\dagger) = m - \operatorname{rank}(A)$.
20. $AA^\dagger = \operatorname{Proj}_{\operatorname{range}(A)}$; $A^\dagger A = \operatorname{Proj}_{\operatorname{range}(A^\dagger)}$.
21. Suppose that $A \in F^{m\times n}$, where $F = \mathbb{C}$ or $\mathbb{R}$. Then
   (a) $\operatorname{range}(A) = \operatorname{range}(AA^*) = \operatorname{range}(AA^\dagger)$.
   (b) $\operatorname{range}(A^\dagger) = \operatorname{range}(A^*) = \operatorname{range}(A^*A) = \operatorname{range}(A^\dagger A)$.
   (c) $\ker(A) = \ker(A^*A) = \ker(A^\dagger A)$.
   (d) $\ker(A^\dagger) = \ker(A^*) = \ker(AA^*) = \ker(AA^\dagger)$.
   (e) $\operatorname{range}(A^\dagger A) \oplus \ker(A^\dagger A) = F^n$.
   (f) $\operatorname{range}(AA^\dagger) \oplus \ker(AA^\dagger) = F^m$.
22. If $A = A_1 + A_2 + \cdots + A_k$, $A_i^*A_j = 0$, and $A_iA_j^* = 0$, for all $i, j = 1, \cdots, k$, $i \neq j$, then $A^\dagger = A_1^\dagger + A_2^\dagger + \cdots + A_k^\dagger$.
23. If A is an $m\times r$ matrix of rank r and B is an $r\times n$ matrix of rank r, then $(AB)^\dagger = B^\dagger A^\dagger$.
24. $(A^*A)^\dagger = A^\dagger(A^*)^\dagger$; $(AA^*)^\dagger = (A^*)^\dagger A^\dagger$.
25. [Gre66] Each one of the following conditions is necessary and sufficient for $(AB)^\dagger = B^\dagger A^\dagger$:
   (a) $\operatorname{range}(BB^*A^*) \subseteq \operatorname{range}(A^*)$ and $\operatorname{range}(A^*AB) \subseteq \operatorname{range}(B)$.
   (b) $A^\dagger ABB^*$ and $A^*ABB^\dagger$ are both Hermitian matrices.
   (c) $A^\dagger ABB^*A^* = BB^*A^*$ and $BB^\dagger A^*AB = A^*AB$.
   (d) $A^\dagger ABB^*A^*ABB^\dagger = BB^*A^*A$.
   (e) $A^\dagger AB = B(AB)^\dagger AB$ and $BB^\dagger A^* = A^*AB(AB)^\dagger$.
26. $(A\otimes B)^\dagger = A^\dagger \otimes B^\dagger$, where $\otimes$ denotes the Kronecker product.
27. $A^\dagger = \lim_{\alpha\to 0} A^*(\alpha I + AA^*)^{-1} = \lim_{\alpha\to 0}(\alpha I + A^*A)^{-1}A^*$.
28. $A^\dagger = \sum_{j=1}^{\infty} A^*(I + AA^*)^{-j} = \sum_{j=1}^{\infty}(I + A^*A)^{-j}A^*$.
29. (Continuity of pseudo-inverse) Suppose that $A \in F^{m\times n}$ and $E \in F^{m\times n}$, where $F = \mathbb{C}$ or $\mathbb{R}$. Then $\lim_{E\to 0}(A+E)^\dagger = A^\dagger$ if and only if there is $\epsilon > 0$ such that $\operatorname{rank}(A+E) = \operatorname{rank}(A)$ when $\|E\|_2 \le \epsilon$.
30. Let $A \in \mathbb{C}^{m\times n}$ be of rank r where $0 < r < \min\{m,n\}$. Suppose that A can be partitioned as
$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$
where $A_{11} \in \mathbb{C}^{r\times r}$ and $\operatorname{rank}(A_{11}) = r$. Then
$$A^\dagger = \begin{bmatrix} A_{11}^*XA_{11}^* & A_{11}^*XA_{21}^* \\ A_{12}^*XA_{11}^* & A_{12}^*XA_{21}^* \end{bmatrix},$$
where
$$X = (A_{11}A_{11}^* + A_{12}A_{12}^*)^{-1}A_{11}(A_{11}^*A_{11} + A_{21}^*A_{21})^{-1}.$$

Noncommutative polynomials

$$\text{Noncommutative polynomials} \quad = \quad \text{elements in free algebra } R\langle X\rangle$$

$$= \quad \sum_{i=1}^{d} c_i \cdot \quad x_{i,1} \ldots x_{i,k_i}$$

Noncommutative polynomials

Noncommutative polynomials $=$ elements in free algebra $R\langle X \rangle$

$$= \sum_{i=1}^{d} c_i \cdot x_{i,1} \dots x_{i,k_i}$$

finite words over X

Noncommutative polynomials

Noncommutative polynomials $=$ elements in free algebra $R\langle X\rangle$

$$= \sum_{i=1}^{d} c_i \cdot \boxed{x_{i,1} \dots x_{i,k_i}}$$

finite words over X

Multiplication $=$ Concatenation of words

$$(x_1 \dots x_k) \cdot (x'_1 \dots x'_l) = x_1 \dots x_k x'_1 \dots x'_l$$

Example: $(ab - 1) \cdot (ba + 1) = abba + ab - ba - 1$

Noncommutative polynomials

Noncommutative polynomials $=$ elements in free algebra $R\langle X\rangle$

$$= \sum_{i=1}^{d} c_i \cdot x_{i,1} \dots x_{i,k_i}$$

finite words over X

Multiplication $=$ Concatenation of words

$$(x_1 \dots x_k) \cdot (x_1' \dots x_l') = x_1 \dots x_k x_1' \dots x_l'$$

Example: $(ab - 1) \cdot (ba + 1) = abba + ab - ba - 1$

Two-sided ideals For $f_1, \dots, f_r \in R\langle X\rangle$

$$(f_1, \dots, f_r) = \left\{ \sum_{i,j} a_{i,j} \cdot f_i \cdot b_{i,j} \;\middle|\; a_{i,j}, b_{i,j} \in R\langle X\rangle \right\}$$

Noncommutative polynomials

Noncommutative polynomials $=$ elements in free algebra $R\langle X\rangle$

$$= \sum_{i=1}^{d} c_i \cdot \boxed{x_{i,1} \dots x_{i,k_i}}$$

finite words over X

Multiplication $=$ Concatenation of words

$$(x_1 \dots x_k) \cdot (x_1' \dots x_l') = x_1 \dots x_k x_1' \dots x_l'$$

Example: $(ab - 1) \cdot (ba + 1) = abba + ab - ba - 1$

Two-sided ideals For $f_1, \dots, f_r \in R\langle X\rangle$

$$(f_1, \dots, f_r) = \left\{ \sum_{i,j} a_{i,j} \cdot f_i \cdot b_{i,j} \;\middle|\; a_{i,j}, b_{i,j} \in R\langle X\rangle \right\}$$

Fact Ideal membership problem $f \overset{?}{\in} (f_1, \dots, f_r)$ is semi-decidable
(e.g., using Gröbner bases)

Operator statements

Operators

- $0, A, B, C, \ldots$
- $S + T, \; S \cdot T, \; f(T_1, \ldots, T_n)$

Linearity = abelian (partial) addition + assoc. (partial) mult. + dist.

Operator statements

Operators

$^*,\ \cdot^{\mathsf{T}},\ \|\cdot\|,\ \otimes, \ldots$

- $0, A, B, C, \ldots$
- $S + T,\ \ S \cdot T,\ f(T_1, \ldots, T_n)$

Linearity = abelian (partial) addition + assoc. (partial) mult. + dist.

Operator statements

Operators

*, $\cdot^\mathsf{T}$, $\|\cdot\|$, $\otimes$, $\ldots$

- $0, A, B, C, \ldots$
- $S + T$, $S \cdot T$, $f(T_1, \ldots, T_n)$

Linearity = abelian (partial) addition + assoc. (partial) mult. + dist.

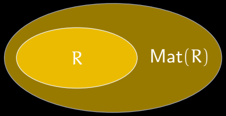
Operator statements

Operators

$^*, \cdot^\mathsf{T}, \|\cdot\|, \otimes, \ldots$

- $0, A, B, C, \ldots$
- $S + T, \; S \cdot T, \; f(T_1, \ldots, T_n)$

Linearity = abelian (partial) addition + assoc. (partial) mult. + dist.



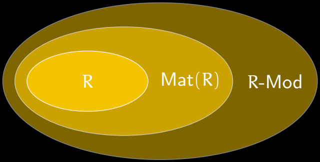
Operator statements

Operators

$^*,\ \cdot^{\mathsf{T}},\ \|\cdot\|,\ \otimes,\ \ldots$

- $0, A, B, C, \ldots$
- $S + T,\ \ S \cdot T,\ \ f(T_1, \ldots, T_n)$

Linearity = abelian (partial) addition + assoc. (partial) mult. + dist.



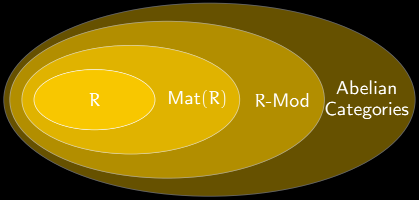
Operator statements

Operators

$*,\ \cdot^{\mathsf{T}},\ \|\cdot\|,\ \otimes, \ldots$

- $0, A, B, C, \ldots$
- $S + T,\ S \cdot T,\ f(T_1, \ldots, T_n)$

Linearity = abelian (partial) addition + assoc. (partial) mult. + dist.



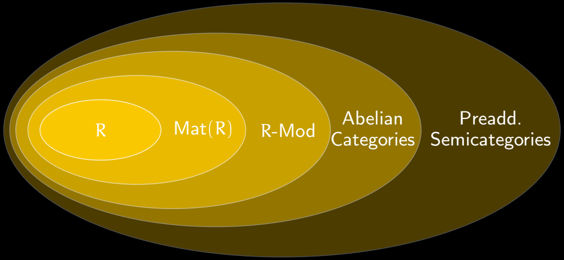
Operator statements

Operators

$*, \cdot^{\mathsf{T}}, \|\cdot\|, \otimes, \ldots$

- $0, A, B, C, \ldots$
- $S + T, \quad S \cdot T, \quad f(T_1, \ldots, T_n)$

Linearity = abelian (partial) addition + assoc. (partial) mult. + dist.



Operator statements

Operators

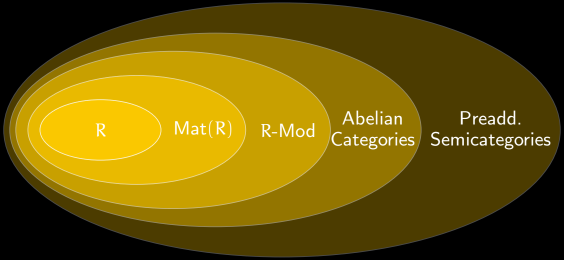
$*,\ \cdot^{\mathsf{T}},\ \|\cdot\|,\ \otimes, \dots$

- $0, A, B, C, \dots$
- $S + T,\ \ S \cdot T,\ \ f(T_1, \dots, T_n)$

Linearity = abelian (partial) addition + assoc. (partial) mult. + dist.

Operator statements

$S = T, \quad \neg\, \varphi, \quad (\varphi \wedge \psi), \quad (\varphi \vee \psi), \quad (\varphi \Rightarrow \psi), \quad \exists X : \varphi, \quad \forall X : \varphi$



Operator statements

Operators

$*,\ \cdot^{\mathsf{T}},\ \|\cdot\|,\ \otimes,\ \ldots$

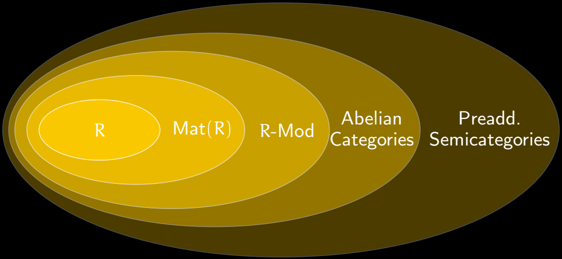
- $0, A, B, C, \ldots$
- $S + T,\ \ S \cdot T,\ \ f(T_1, \ldots, T_n)$

Linearity = abelian (partial) addition + assoc. (partial) mult. + dist.

Operator statements

$S = T,\quad \neg\,\varphi,\quad (\varphi \wedge \psi),\quad (\varphi \vee \psi),\quad (\varphi \Rightarrow \psi),\quad \exists X : \varphi,\quad \forall X : \varphi$

Definition An operator statement is universally true if it follows from linearity



Operator statements

Operators

$*,\ \cdot^{\mathsf{T}},\ \|\cdot\|,\ \otimes, \dots$

- $0, A, B, C, \dots$
- $S + T,\ S \cdot T,\ f(T_1, \dots, T_n)$

Linearity = abelian (partial) addition + assoc. (partial) mult. + dist.

Operator statements

$S = T, \quad \neg\, \varphi, \quad (\varphi \wedge \psi), \quad (\varphi \vee \psi), \quad (\varphi \Rightarrow \psi), \quad \exists X : \varphi, \quad \forall X : \varphi$

Definition An operator statement is universally true if it follows from linearity

- Fact: Determining universal truth is not decidable
  $\Rightarrow$ Algorithm that terminates on all inputs cannot exist

Operator statements

Operators

$*, \cdot^T, \|\cdot\|, \otimes, \ldots$

- $0, A, B, C, \ldots$
- $S + T,\ S \cdot T,\ f(T_1, \ldots, T_n)$

Linearity = abelian (partial) addition + assoc. (partial) mult. + dist.

Operator statements

$$S = T, \quad \neg\varphi, \quad (\varphi \wedge \psi), \quad (\varphi \vee \psi), \quad (\varphi \Rightarrow \psi), \quad \exists X : \varphi, \quad \forall X : \varphi$$

Definition An operator statement is universally true if it follows from linearity

- Fact: Determining universal truth is not decidable
  $\Rightarrow$ Algorithm that terminates on all inputs cannot exist
- Best we can hope for: (effective) semi-decision procedure
  $\rightarrow$ Can be obtained using computer algebra

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$ABA = A, \qquad BAB = B, \qquad B^*A^* = AB, \qquad A^*B^* = BA$$

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$ABA = A, \qquad BAB = B, \qquad B^*A^* = AB, \qquad A^*B^* = BA$$

Claim $\quad \forall A, B, C \; : \; \mathsf{mp}(A, B) \; \wedge \; \mathsf{mp}(A, C) \quad \Rightarrow \quad B = C$

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$ABA = A, \qquad BAB = B, \qquad B^*A^* = AB, \qquad A^*B^* = BA$$

Claim $\quad \forall A, B, C \ : \ \mathrm{mp}(A, B) \ \wedge \ \mathrm{mp}(A, C) \quad \Rightarrow \quad B = C$

Proof $\quad B \ = \ BAB \ = \ BACAB \ = \ \ldots \ = \ C$

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$ABA = A, \qquad BAB = B, \qquad B^*A^* = AB, \qquad A^*B^* = BA$$

Claim $\quad \forall A, B, C \;:\; \mathsf{mp}(A, B) \,\wedge\, \mathsf{mp}(A, C) \;\Rightarrow\; B = C$

Proof $\quad B = BAB = BACAB = \ldots = C$

From identities to polynomials

$$L = R \iff L - R = 0$$

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$ABA = A, \qquad BAB = B, \qquad B^*A^* = AB, \qquad A^*B^* = BA$$

Claim $\quad \forall A, B, C \, : \, \mathsf{mp}(A, B) \wedge \mathsf{mp}(A, C) \Rightarrow B = C$

Proof $\quad B = BAB = BACAB = \ldots = C$

From identities to polynomials

$$L = R \iff L - R$$

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$ABA = A, \qquad BAB = B, \qquad B^*A^* = AB, \qquad A^*B^* = BA$$

Claim $\quad \forall A, B, C \; : \; \mathsf{mp}(A, B) \; \wedge \; \mathsf{mp}(A, C) \;\; \Rightarrow \;\; B = C$

Proof $\quad B \; = \; BAB \; = \; BACAB \; = \; \ldots \; = \; C$

From identities to polynomials

$$L = R \qquad \Longleftrightarrow \qquad l - r \; \in \; \mathbb{Z}\langle \mathbf{X} \rangle$$

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$aba - a, \qquad bab - b, \qquad b^*a^* - ab, \qquad a^*b^* - ba$$

Claim $\forall A, B, C \; : \; \text{mp}(A, B) \wedge \text{mp}(A, C) \;\Rightarrow\; B = C$

Proof $B = BAB = BACAB = \ldots = C$

From identities to polynomials

$$L = R \quad \Longleftrightarrow \quad l - r \in \mathbb{Z}\langle \mathbf{X} \rangle$$

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$aba - a, \qquad bab - b, \qquad b^*a^* - ab, \qquad a^*b^* - ba$$

Claim $\forall A, B, C : \mathrm{mp}(A, B) \wedge \mathrm{mp}(A, C) \Rightarrow B = C$

Proof $B = BAB = BACAB = \ldots = C$

From identities to polynomials

$$L = R \iff l - r \in \mathbb{Z}\langle \mathbf{X} \rangle$$

$$B = \ldots = C \iff b - c \in (f_1, \ldots, f_{12})$$

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$aba - a, \qquad bab - b, \qquad b^*a^* - ab, \qquad a^*b^* - ba$$

Claim $\forall A, B, C : \mathrm{mp}(A, B) \wedge \mathrm{mp}(A, C) \Rightarrow B = C$

Proof $B = BAB = BACAB = \ldots = C$

From identities to polynomials

$$L = R \iff l - r \in \mathbb{Z}\langle \mathbf{X} \rangle$$

$$B = \ldots = C \iff b - c \in (f_1, \ldots, f_{12})$$

Theorem (Helton, Stankus, Wavrik '98, Schmitz, Levandovskyy '20, Raab, Regensburger, Hossein Poor '21)

$$\forall \mathbf{X} : \bigwedge_{i=1}^{m} P_i = Q_i \Rightarrow S = T \quad \text{iff} \quad s - t \in (p_1 - q_1, \ldots, p_m - q_m)$$

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$aba - a, \qquad bab - b, \qquad b^* a^* - ab, \qquad a^* b^* - ba$$

Claim $\forall A, B, C : \text{mp}(A, B) \wedge \text{mp}(A, C) \Rightarrow B = C$

Proof $B = BAB = BACAB = \ldots = C$

From identities to polynomials

$$L = R \iff l - r \in \mathbb{Z}\langle \mathbf{X} \rangle$$

$$B = \ldots = C \iff b - c \in (f_1, \ldots, f_{12})$$

Theorem (Helton, Stankus, Wavrik '98, Schmitz, Levandovskyy '20, Raab, Regensburger, Hossein Poor '21)

$\forall \mathbf{X} : \bigwedge_{i=1}^{m} P_i = Q_i \Rightarrow S = T$ iff

$$s - t = \sum_{i,j} a_{i,j} \cdot (p_i - q_i) \cdot b_{i,j}$$

- *"cofactor representation"*
- computable with Gröbner bases

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$aba - a, \qquad bab - b, \qquad b^*a^* - ab, \qquad a^*b^* - ba$$

Claim $\quad \forall A, B, C \,:\, \mathsf{mp}(A,B) \wedge \mathsf{mp}(A,C) \;\Rightarrow\; B = C$

Proof Using our software package `operator_gb`...

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$aba - a, \qquad bab - b, \qquad b^*a^* - ab, \qquad a^*b^* - ba$$

Claim $\forall A, B, C : \mathrm{mp}(A, B) \wedge \mathrm{mp}(A, C) \Rightarrow B = C$

Proof Using our software package operator_gb...

```
sage: from operator_gb import *
sage: assumptions = [a*b*a - a,...]
sage: certify(assumptions, b - c)
```

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$aba - a, \qquad bab - b, \qquad b^*a^* - ab, \qquad a^*b^* - ba$$

Claim $\forall A, B, C : \text{mp}(A, B) \wedge \text{mp}(A, C) \Rightarrow B = C$

Proof Using our software package operator_gb...

```
sage: from operator_gb import *
sage: assumptions = [a*b*a - a,...]
sage: certify(assumptions, b - c)
b - c = (-c + c*a*c) + b*c_adj*(-a_adj + a_adj*b_adj*a_adj)
        - b*a*c*(-a*b + b_adj*a_adj) - b*(-a + a*c*a)*b
        + b*(-a*c + c_adj*a_adj) - b*(-a*c + c_adj*a_adj)*b_adj*a_adj
        - (-b + b*a*b) + (-c*a + a_adj*c_adj)*b*a*c
        - (-a_adj + a_adj*c_adj*a_adj)*b_adj*c + c*(-a + a*b*a)*c
        - (-b*a + a_adj*b_adj)*c + a_adj*c_adj*(-b*a + a_adj*b_adj)*c
```

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

$$aba - a\,, \qquad bab - b\,, \qquad b^*a^* - ab\,, \qquad a^*b^* - ba$$

Claim $\forall A, B, C\ :\ \mathrm{mp}(A,B) \wedge \mathrm{mp}(A,C) \Rightarrow B = C$

Proof Using our software package operator_gb...

```
sage: from operator_gb import *
sage: assumptions = [a*b*a - a,...]
sage: certify(assumptions, b - c)
b - c = (-c + c*a*c) + b*c_adj*(-a_adj + a_adj*b_adj*a_adj)
        - b*a*c*(-a*b + b_adj*a_adj) - b*(-a + a*c*a)*b
        + b*(-a*c + c_adj*a_adj) - b*(-a*c + c_adj*a_adj)*b_adj*a_adj
        - (-b + b*a*b) + (-c*a + a_adj*c_adj)*b*a*c
        - (-a_adj + a_adj*c_adj*a_adj)*b_adj*c + c*(-a + a*b*a)*c
        - (-b*a + a_adj*b_adj)*c + a_adj*c_adj*(-b*a + a_adj*b_adj)*c
```

- Software produces cofactor representation (= algebraic proof)

$\Rightarrow$ Operator statement is universally true

Determining universal truth

Quasi-identities

(Helton, Stankus, Wavrik '98, Schmitz, Levandovskyy '20, Raab, Regensburger, Hossein Poor '21)

$$\forall \mathbf{X} : \bigwedge_{i=1}^{m} P_i = Q_i \;\Rightarrow\; S = T \qquad \text{iff} \qquad s - t \in (p_1 - q_1, \ldots, p_m - q_m)$$

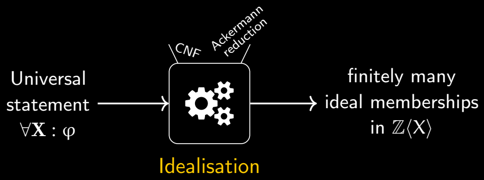
Determining universal truth

Universal statements

Universal
statement
$\forall \mathbf{X} : \varphi$

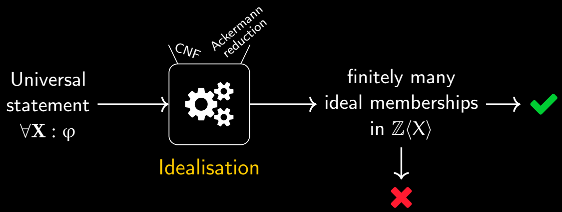
Determining universal truth

Universal statements



Determining universal truth

Universal statements



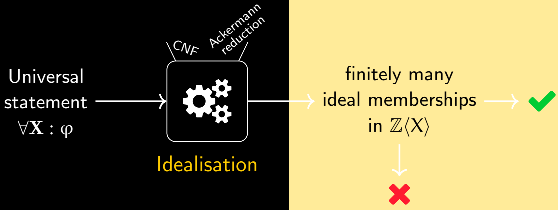
Theorem (H., Raab, Regensburger '22b)
A universal statement is universally true iff its idealisation is true

H., Raab, Regensburger. *Universal truth of operator statements via ideal membership*. preprint. 2022.

Determining universal truth

Universal statements

efficiently verifiable using Gröbner bases



Theorem (H., Raab, Regensburger '22b)
A universal statement is universally true iff its idealisation is true

---

H., Raab, Regensburger. *Universal truth of operator statements via ideal membership*. preprint. 2022.

5.7 Pseudo-Inverse

Definitions:

A **Moore–Penrose pseudo-inverse** of a matrix $A \in \mathbb{C}^{m\times n}$ is a matrix $A^\dagger \in \mathbb{C}^{n\times m}$ that satisfies the following four **Penrose** conditions:

$$AA^\dagger A = A; \quad A^\dagger AA^\dagger = A^\dagger; \quad (AA^\dagger)^* = AA^\dagger; \quad (A^\dagger A)^* = A^\dagger A.$$

Facts:

All the following facts except those with a specific reference can be found in [Gra83, pp. 105–141] or [RM71, pp. 44–67].

1. Every $A \in \mathbb{C}^{m\times n}$ has a unique pseudo-inverse $A^\dagger$.
2. If $A \in \mathbb{R}^{m\times n}$, then $A^\dagger$ is real.
3. If $A \in \mathbb{C}^{m\times n}$ of rank r has a full rank decomposition $A = BC$, where $B \in \mathbb{C}^{m\times r}$ and $C \in \mathbb{C}^{r\times n}$, then $A^\dagger$ can be evaluated using $A^\dagger = C^*(B^*AC^*)^{-1}B^*$.
4. [LH95, p. 38] If $A \in \mathbb{C}^{m\times n}$ of rank $r \le \min\{m,n\}$ has an SVD $A = U\Sigma V^*$, then its pseudo-inverse is $A^\dagger = V\Sigma^\dagger U^*$, where
$$\Sigma^\dagger = \operatorname{diag}(1/\sigma_1, \dots, 1/\sigma_r, 0, \dots, 0) \in \mathbb{R}^{n\times m}.$$
5. [Hig96, p. 412] The pseudo-inverse $A^\dagger$ of $A \in F^{m\times n}$ ($F = \mathbb{C}$ or $\mathbb{R}$) solves the minimization problem
$$\min_{X \in F^{n\times m}} \|AX - I_m\|_F^2.$$
6. $\mathbf{0}_{mn}^\dagger = \mathbf{0}_{nm}$ and $J_{mn}^\dagger = \frac{1}{mn}J_{nm}$, where $\mathbf{0}_{mn} \in \mathbb{C}^{m\times n}$ is the all 0s matrix and $J_{mn} \in \mathbb{C}^{m\times n}$ is the all 1s matrix.
7. If $\mathbf{x} \neq \mathbf{0}$, $\mathbf{y} \neq \mathbf{0}$, then $(\mathbf{x}\mathbf{y}^*)^\dagger = \dfrac{\mathbf{y}\mathbf{x}^*}{\|\mathbf{x}\|^2\|\mathbf{y}\|^2}$.
8. If $\mathbf{x} \neq \mathbf{0}$, then $\mathbf{x}^\dagger = \dfrac{\mathbf{x}^*}{\|\mathbf{x}\|^2}$.
9. Let α be a scalar. Denote
$$\alpha^\dagger = \begin{cases} \alpha^{-1}, & \text{if } \alpha \neq 0, \\ 0, & \text{if } \alpha = 0. \end{cases}$$
Then
   (a) $(\alpha A)^\dagger = \alpha^\dagger A^\dagger$.
   (b) $(\operatorname{diag}(\beta_1, \beta_2, \cdots, \beta_n))^\dagger = \operatorname{diag}(\beta_1^\dagger, \beta_2^\dagger, \cdots, \beta_n^\dagger)$.
10. $(A^\dagger)^* = (A^*)^\dagger$; $(A^\dagger)^\dagger = A$.
11. If A is a nonsingular square matrix, then $A^\dagger = A^{-1}$.
12. If U has orthonormal columns or orthonormal rows, then $U^\dagger = U^*$.
13. If $A = A^*$ and $A = A^2$, then $A^\dagger = A$.
14. $A^\dagger = A^*$ if and only if A^*A is idempotent.
15. If A is normal and k is a positive integer, then $AA^\dagger = A^\dagger A$ and $(A^k)^\dagger = (A^\dagger)^k$.
16. If $U \in \mathbb{C}^{m\times n}$ is of rank n and satisfies $U^\dagger = U^*$, then U has orthonormal columns.
17. If $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are unitary matrices, then $(UAV)^\dagger = V^*A^\dagger U^*$.
18. $A^\dagger = (A^*A)^\dagger A^* = A^*(AA^*)^\dagger$. In particular,
   (a) if $A \in \mathbb{C}^{m\times n}$ $(m \ge n)$ has full rank n, then $A^\dagger = (A^*A)^{-1}A^*$;
   (b) if $A \in \mathbb{C}^{m\times n}$ $(m \le n)$ has full rank m, then $A^\dagger = A^*(AA^*)^{-1}$.
19. Let $A \in \mathbb{C}^{m\times n}$. Then

   (a) $A^\dagger A$, $AA^\dagger$, $I_n - A^\dagger A$, and $I_m - AA^\dagger$ are orthogonal projections.
   (b) $\operatorname{rank}(A) = \operatorname{rank}(A^\dagger) = \operatorname{rank}(AA^\dagger) = \operatorname{rank}(A^\dagger A)$.
   (c) $\operatorname{rank}(I_n - A^\dagger A) = n - \operatorname{rank}(A)$.
   (d) $\operatorname{rank}(I_m - AA^\dagger) = m - \operatorname{rank}(A)$.
20. $AA^\dagger = \operatorname{Proj}_{\operatorname{range}(A)}$; $A^\dagger A = \operatorname{Proj}_{\operatorname{range}(A^\dagger)}$.
21. Suppose that $A \in F^{m\times n}$, where $F = \mathbb{C}$ or $\mathbb{R}$. Then
   (a) $\operatorname{range}(A) = \operatorname{range}(AA^*) = \operatorname{range}(AA^\dagger)$.
   (b) $\operatorname{range}(A^\dagger) = \operatorname{range}(A^*) = \operatorname{range}(A^*A) = \operatorname{range}(A^\dagger A)$.
   (c) $\ker(A) = \ker(A^*A) = \ker(A^\dagger A)$.
   (d) $\ker(A^\dagger) = \ker(A^*) = \ker(AA^*) = \ker(AA^\dagger)$.
   (e) $\operatorname{range}(A^\dagger A) \oplus \ker(A^\dagger A) = F^n$.
   (f) $\operatorname{range}(AA^\dagger) \oplus \ker(AA^\dagger) = F^m$.
22. If $A = A_1 + A_2 + \cdots + A_k$, $A_i^*A_j = 0$, and $A_iA_j^* = 0$, for all $i, j = 1, \cdots, k$, $i \neq j$, then $A^\dagger = A_1^\dagger + A_2^\dagger + \cdots + A_k^\dagger$.
23. If A is an $m\times r$ matrix of rank r and B is an $r\times n$ matrix of rank r, then $(AB)^\dagger = B^\dagger A^\dagger$.
24. $(A^*A)^\dagger = A^\dagger(A^*)^\dagger$; $(AA^*)^\dagger = (A^*)^\dagger A^\dagger$.
25. [Gre66] Each one of the following conditions is necessary and sufficient for $(AB)^\dagger = B^\dagger A^\dagger$:
   (a) $\operatorname{range}(BB^*A^*) \subseteq \operatorname{range}(A^*)$ and $\operatorname{range}(A^*AB) \subseteq \operatorname{range}(B)$.
   (b) $A^\dagger ABB^*$ and $A^*ABB^\dagger$ are both Hermitian matrices.
   (c) $A^\dagger ABB^*A^* = BB^*A^*$ and $BB^\dagger A^*AB = A^*AB$.
   (d) $A^\dagger ABB^*A^*ABB^\dagger = BB^*A^*A$.
   (e) $A^\dagger AB = B(AB)^\dagger AB$ and $BB^\dagger A^* = A^*AB(AB)^\dagger$.
26. $(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$, where $\otimes$ denotes the Kronecker product.
27. $A^\dagger = \lim\limits_{\alpha\to 0} A^*(\alpha I + AA^*)^{-1} = \lim\limits_{\alpha\to 0}(\alpha I + A^*A)^{-1}A^*$.
28. $A^\dagger = \sum\limits_{j=1}^{\infty} A^*(I + AA^*)^{-j} = \sum\limits_{j=1}^{\infty}(I + A^*A)^{-j}A^*$.
29. (Continuity of pseudo-inverse) Suppose that $A \in F^{m\times n}$ and $E \in F^{m\times n}$, where $F = \mathbb{C}$ or $\mathbb{R}$. Then $\lim\limits_{E\to 0}(A + E)^\dagger = A^\dagger$ if and only if there is $\epsilon > 0$ such that $\operatorname{rank}(A+E) = \operatorname{rank}(A)$ when $\|E\|_2 \le \epsilon$.
30. Let $A \in \mathbb{C}^{m\times n}$ be of rank r where $0 < r < \min\{m,n\}$. Suppose that A can be partitioned as
$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$
where $A_{11} \in \mathbb{C}^{r\times r}$ and $\operatorname{rank}(A_{11}) = r$. Then
$$A^\dagger = \begin{bmatrix} A_{11}^*XA_{11}^* & A_{11}^*XA_{21}^* \\ A_{12}^*XA_{11}^* & A_{12}^*XA_{21}^* \end{bmatrix},$$
where
$$X = (A_{11}A_{11}^* + A_{12}A_{12}^*)^{-1}A_{11}(A_{11}^*A_{11} + A_{21}^*A_{21})^{-1}.$$

“Every matrix has a Moore-Penrose inverse”

“*Every matrix* *has a Moore-Penrose inverse*”

"Every matrix has a Moore-Penrose inverse"

Fact: A matrix $\quad\Rightarrow\quad \exists\, P, Q \;:\; PA^*A = A$ and $AA^*Q = A$

"Every matrix has a Moore-Penrose inverse"

Fact: A matrix $\Rightarrow$ $\exists\, P, Q \;:\; PA^*A = A$ and $AA^*Q = A$

Claim $\exists X : (PA^*A = A \wedge AA^*Q = A) \;\Rightarrow\; \mathrm{mp}(A, X)$

"Every matrix has a Moore-Penrose inverse"

Fact: A matrix $\Rightarrow$ $\exists\, P, Q \;:\; PA^*A = A$ and $AA^*Q = A$

Claim $\exists X : (PA^*A = A \wedge AA^*Q = A) \;\Rightarrow\; \mathrm{mp}(A, X)$

Strategy

1. Derive explicit expression for X
2. Reformulate statement as a universal statement
3. Prove by computing idealisation

"Every matrix has a Moore-Penrose inverse"

Fact: A matrix $\quad \Rightarrow \quad \exists\, P, Q \;:\; PA^*A = A$ and $AA^*Q = A$

Claim $\quad \exists X : (PA^*A = A \wedge AA^*Q = A) \;\Rightarrow\; \mathrm{mp}(A, X)$

Strategy

1. Derive explicit expression for X
2. Reformulate statement as a universal statement
3. Prove by computing idealisation

Proof Using our software package `operator_gb`...

"Every matrix has a Moore-Penrose inverse"

Fact: A matrix $\Rightarrow$ $\exists\, P, Q\ :\ PA^*A = A$ and $AA^*Q = A$

Claim $\exists X : (PA^*A = A \wedge AA^*Q = A) \Rightarrow \mathrm{mp}(A, X)$

Strategy

1. Derive explicit expression for X
2. Reformulate statement as a universal statement
3. Prove by computing idealisation

Proof Using our software package operator_gb...

```
sage: assumptions =  [a - p*a_adj*a,...]
sage: I = NCIdeal(assumptions + pinv(a,x))
sage: I.find_equivalent_expression(x)
```

"Every matrix has a Moore-Penrose inverse"

Fact: A matrix $\Rightarrow$ $\exists\, P, Q \,:\, PA^*A = A$ and $AA^*Q = A$

Claim $\exists X : (PA^*A = A \wedge AA^*Q = A) \Rightarrow \mathrm{mp}(A, X)$

Strategy

1. Derive explicit expression for X
2. Reformulate statement as a universal statement
3. Prove by computing idealisation

Proof Using our software package operator_gb...

```
sage: assumptions =  [a - p*a_adj*a,...]
sage: I = NCIdeal(assumptions + pinv(a,x))
sage: I.find_equivalent_expression(x)
```

```
[x - a_adj*q*x, x - a_adj*p*x,
          x - a_adj*q*p_adj, x - a_adj*x_adj*x]
```

"Every matrix has a Moore-Penrose inverse"

Fact: A matrix $\Rightarrow$ $\exists\, P, Q \;:\; PA^*A = A$ and $AA^*Q = A$

Claim $\exists X : (PA^*A = A \wedge AA^*Q = A) \Rightarrow \mathrm{mp}(A, X)$

Strategy

1. Derive explicit expression for X
2. Reformulate statement as a universal statement
3. Prove by computing idealisation

Proof Using our software package operator_gb...

```
sage: assumptions =  [a - p*a_adj*a,...]
sage: I = NCIdeal(assumptions + pinv(a,x))
sage: I.find_equivalent_expression(x)
```

```
[x - a_adj*q*x, x - a_adj*p*x,
          x - a_adj*q*p_adj, x - a_adj*x_adj*x]
```

"Every matrix has a Moore-Penrose inverse"

Fact: A matrix $\Rightarrow$ $\exists\, P, Q \;:\; PA^*A = A$ and $AA^*Q = A$

Claim $\exists X : (PA^*A = A \wedge AA^*Q = A) \;\Rightarrow\; \mathrm{mp}(A, X)$

Strategy

1. Derive explicit expression for X
2. Reformulate statement as a universal statement
3. Prove by computing idealisation

Proof Using our software package operator_gb...

```
sage: assumptions =  [a - p*a_adj*a,...]
sage: I = NCIdeal(assumptions + pinv(a,x))
sage: I.find_equivalent_expression(x)
      [x - a_adj*q*x, x - a_adj*p*x,
                x - a_adj*q*p_adj, x - a_adj*x_adj*x]
```

$\Rightarrow$ $X = A^*QP^*$ is MP-inverse of A

(can be certified using the software)

Existential statements

In the previous example, we found a suitable polynomial expression.

Question Was this just luck?

Existential statements

In the previous example, we found a suitable polynomial expression.

Question Was this just luck?

Answer No! – Herbrand's theorem (Herbrand '30)

Such expressions always exist and the possible candidates are enumerable.

Existential statements

In the previous example, we found a suitable polynomial expression.

Question Was this just luck?

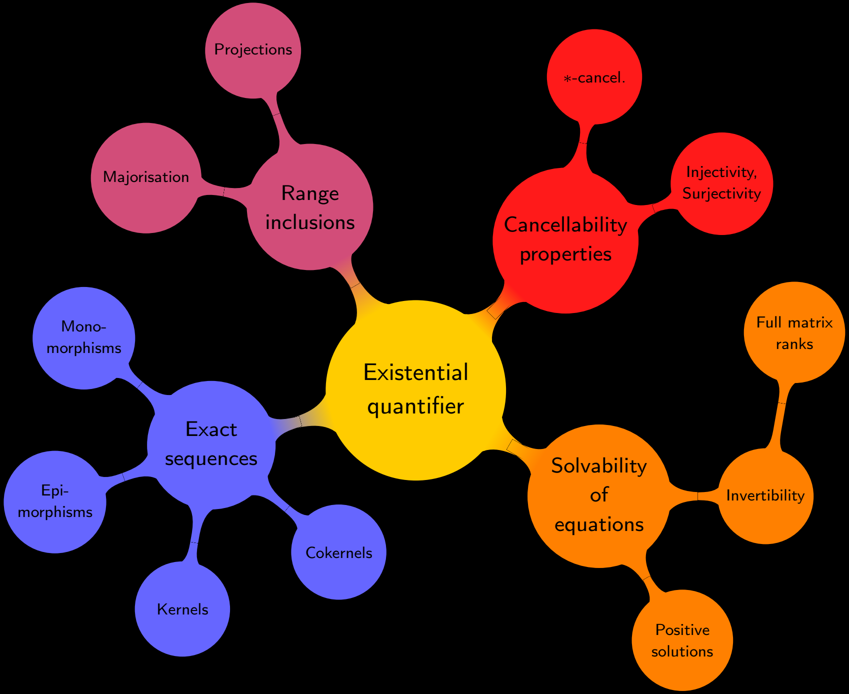
Answer No! – Herbrand's theorem (Herbrand '30)

Such expressions always exist and the possible candidates are enumerable.

- Enumerating all possible expressions is hopeless
- Requires good heuristics $\longrightarrow$ provided by computer algebra
  (H., Raab, Regensburger '22a)
- Several heuristics implemented in `operator_gb`
  (ansatz, ideal/subalgebra intersections, hom. part, monomial part,...)

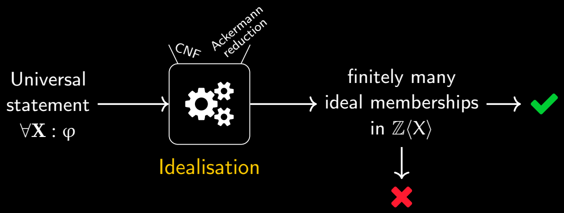
H., Raab, Regensburger. *Computing Elements of Certain Form in Ideals to Prove Properties of Operators*. In: Mathematics in Computer Science. 2022.

Existential
quantifier


Existential quantifier
Range inclusions
Projections
Majorisation
Cancellability properties
∗-cancel.
Injectivity, Surjectivity
Solvability of equations
Invertibility
Full matrix ranks
Positive solutions
Exact sequences
Mono-morphisms
Epi-morphisms
Kernels
Cokernels

Determining universal truth

Universal statements

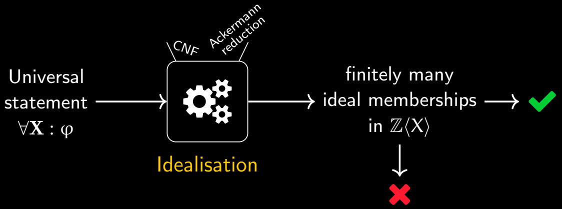


Theorem (H., Raab, Regensburger '22b)
A universal statement is universally true iff its idealisation is true

H., Raab, Regensburger. *Universal truth of operator statements via ideal membership*. preprint. 2022.

Determining universal truth

Universal statements



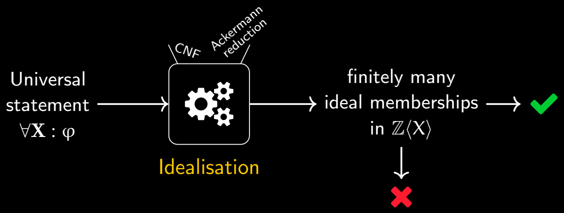
Theorem (H., Raab, Regensburger '22b)
A universal statement is universally true iff its idealisation is true

To treat all operator statements $\rightsquigarrow$ combine with Herbrand's theorem
+ Heuristics

H., Raab, Regensburger. *Universal truth of operator statements via ideal membership*. preprint. 2022.

Determining universal truth

General operator statements



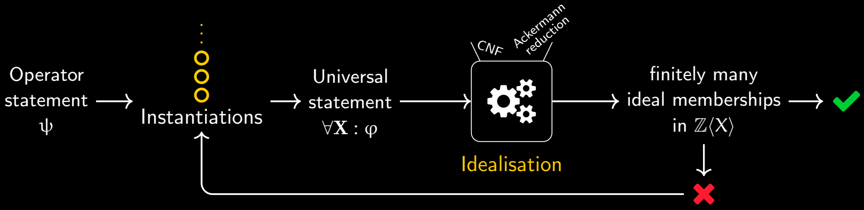
Theorem (H., Raab, Regensburger '22b)
A universal statement is universally true iff its idealisation is true

To treat all operator statements $\rightsquigarrow$ combine with Herbrand's theorem
+ Heuristics

H., Raab, Regensburger. *Universal truth of operator statements via ideal membership*. preprint. 2022.

Determining universal truth

General operator statements



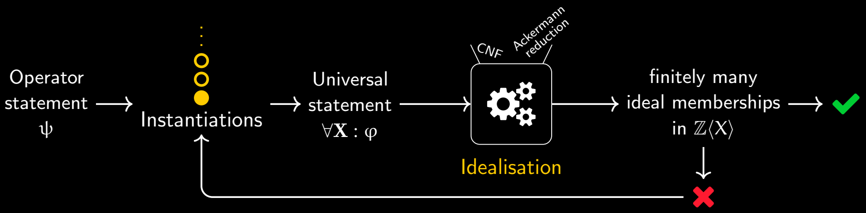
Theorem (H., Raab, Regensburger '22b)
A universal statement is universally true iff its idealisation is true

To treat all operator statements $\rightsquigarrow$ combine with Herbrand's theorem
+ Heuristics

H., Raab, Regensburger. *Universal truth of operator statements via ideal membership*. preprint. 2022.

Determining universal truth

General operator statements



Theorem (H., Raab, Regensburger '22b)
A universal statement is universally true iff its idealisation is true

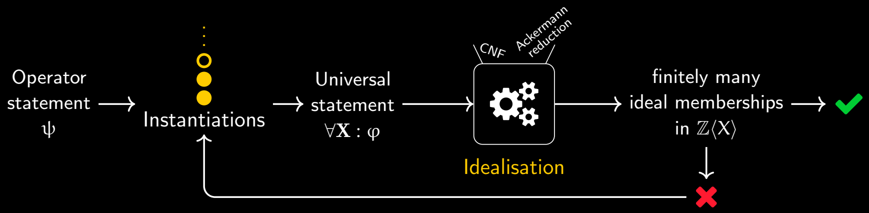
To treat all operator statements $\rightsquigarrow$ combine with Herbrand's theorem + Heuristics

---

H., Raab, Regensburger. *Universal truth of operator statements via ideal membership*. preprint. 2022.

Determining universal truth

General operator statements



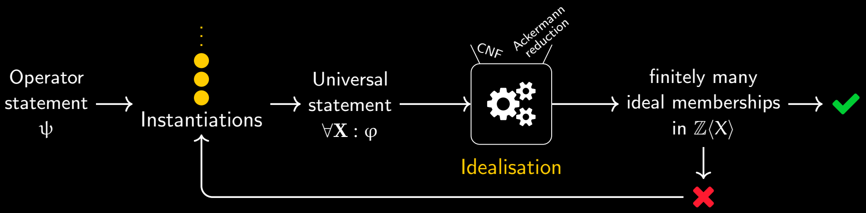
Theorem (H., Raab, Regensburger '22b)
A universal statement is universally true iff its idealisation is true

To treat all operator statements $\rightsquigarrow$ combine with Herbrand's theorem
+ Heuristics

H., Raab, Regensburger. *Universal truth of operator statements via ideal membership*. preprint. 2022.

Determining universal truth

General operator statements



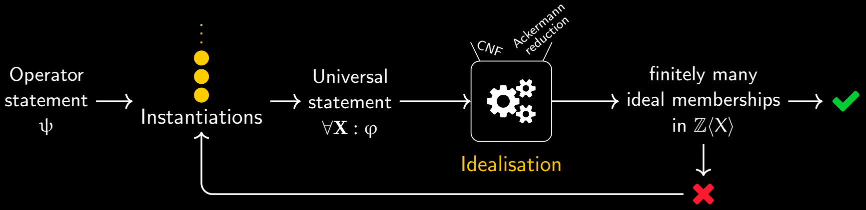
Theorem (H., Raab, Regensburger '22b)
A universal statement is universally true iff its idealisation is true

To treat all operator statements $\rightsquigarrow$ combine with Herbrand's theorem
+ Heuristics

H., Raab, Regensburger. *Universal truth of operator statements via ideal membership*. preprint. 2022.

Determining universal truth

General operator statements



Theorem (H., Raab, Regensburger '22b)
An operator statement is universally true iff the procedure terminates and returns ✔

To treat all operator statements $\rightsquigarrow$ combine with Herbrand's theorem
+ Heuristics

H., Raab, Regensburger. *Universal truth of operator statements via ideal membership*. preprint. 2022.

5.7 Pseudo-Inverse

Definitions:

A **Moore–Penrose pseudo-inverse** of a matrix $A \in \mathbb{C}^{m\times n}$ is a matrix $A^\dagger \in \mathbb{C}^{n\times m}$ that satisfies the following four **Penrose** conditions:

$$AA^\dagger A = A; \quad A^\dagger AA^\dagger = A^\dagger; \quad (AA^\dagger)^* = AA^\dagger; \quad (A^\dagger A)^* = A^\dagger A.$$

Facts:

All the following facts except those with a specific reference can be found in [Gra83, pp. 105–141] or [RM71, pp. 44–67].

1. Every $A \in \mathbb{C}^{m\times n}$ has a unique pseudo-inverse $A^\dagger$.
2. If $A \in \mathbb{R}^{m\times n}$, then $A^\dagger$ is real.
3. If $A \in \mathbb{C}^{m\times n}$ of rank r has a full rank decomposition $A = BC$, where $B \in \mathbb{C}^{m\times r}$ and $C \in \mathbb{C}^{r\times n}$, then $A^\dagger$ can be evaluated using $A^\dagger = C^*(B^*AC^*)^{-1}B^*$.
4. [LH95, p. 38] If $A \in \mathbb{C}^{m\times n}$ of rank $r \le \min\{m,n\}$ has an SVD $A = U\Sigma V^*$, then its pseudo-inverse is $A^\dagger = V\Sigma^\dagger U^*$, where

$$\Sigma^\dagger = \operatorname{diag}(1/\sigma_1, \ldots, 1/\sigma_r, 0, \ldots, 0) \in \mathbb{R}^{n\times m}.$$

5. [Hig96, p. 412] The pseudo-inverse $A^\dagger$ of $A \in F^{m\times n}$ ($F = \mathbb{C}$ or $\mathbb{R}$) solves the minimization problem

$$\min_{X\in F^{n\times m}} \|AX - I_m\|_F^2.$$

6. $\mathbf{0}_{mn}^\dagger = \mathbf{0}_{nm}$ and $J_{mn}^\dagger = \frac{1}{mn}J_{nm}$, where $\mathbf{0}_{mn} \in \mathbb{C}^{m\times n}$ is the all 0s matrix and $J_{mn} \in \mathbb{C}^{m\times n}$ is the all 1s matrix.
7. If $\mathbf{x} \neq \mathbf{0}$, $\mathbf{y} \neq \mathbf{0}$, then $(\mathbf{x}\mathbf{y}^*)^\dagger = \dfrac{\mathbf{y}\mathbf{x}^*}{\|\mathbf{x}\|^2\|\mathbf{y}\|^2}$.
8. If $\mathbf{x} \neq \mathbf{0}$, then $\mathbf{x}^\dagger = \dfrac{\mathbf{x}^*}{\|\mathbf{x}\|^2}$.
9. Let α be a scalar. Denote

$$\alpha^\dagger = \begin{cases}\alpha^{-1}, & \text{if } \alpha \neq 0,\\ 0, & \text{if } \alpha = 0.\end{cases}$$

Then

(a) $(\alpha A)^\dagger = \alpha^\dagger A^\dagger$.

(b) $(\operatorname{diag}(\beta_1, \beta_2, \cdots, \beta_n))^\dagger = \operatorname{diag}(\beta_1^\dagger, \beta_2^\dagger, \cdots, \beta_n^\dagger)$.

10. $(A^\dagger)^* = (A^*)^\dagger$; $(A^\dagger)^\dagger = A$.
11. If A is a nonsingular square matrix, then $A^\dagger = A^{-1}$.
12. If U has orthonormal columns or orthonormal rows, then $U^\dagger = U^*$.
13. If $A = A^*$ and $A = A^2$, then $A^\dagger = A$.
14. $A^\dagger = A^*$ if and only if A^*A is idempotent.
15. If A is normal and k is a positive integer, then $AA^\dagger = A^\dagger A$ and $(A^k)^\dagger = (A^\dagger)^k$.
16. If $U \in \mathbb{C}^{m\times n}$ is of rank n and satisfies $U^\dagger = U^*$, then U has orthonormal columns.
17. If $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are unitary matrices, then $(UAV)^\dagger = V^*A^\dagger U^*$.
18. $A^\dagger = (A^*A)^\dagger A^* = A^*(AA^*)^\dagger$. In particular,

(a) if $A \in \mathbb{C}^{m\times n}$ ($m \ge n$) has full rank n, then $A^\dagger = (A^*A)^{-1}A^*$;

(b) if $A \in \mathbb{C}^{m\times n}$ ($m \le n$) has full rank m, then $A^\dagger = A^*(AA^*)^{-1}$.

19. Let $A \in \mathbb{C}^{m\times n}$. Then

(a) $A^\dagger A$, $AA^\dagger$, $I_n - A^\dagger A$, and $I_m - AA^\dagger$ are orthogonal projections.

(b) $\operatorname{rank}(A) = \operatorname{rank}(A^\dagger) = \operatorname{rank}(AA^\dagger) = \operatorname{rank}(A^\dagger A)$.

(c) $\operatorname{rank}(I_n - A^\dagger A) = n - \operatorname{rank}(A)$.

(d) $\operatorname{rank}(I_m - AA^\dagger) = m - \operatorname{rank}(A)$.

20. $AA^\dagger = \operatorname{Proj}_{\operatorname{range}(A)}$; $A^\dagger A = \operatorname{Proj}_{\operatorname{range}(A^\dagger)}$.
21. Suppose that $A \in F^{m\times n}$, where $F = \mathbb{C}$ or $\mathbb{R}$. Then

(a) $\operatorname{range}(A) = \operatorname{range}(AA^*) = \operatorname{range}(AA^\dagger)$.

(b) $\operatorname{range}(A^\dagger) = \operatorname{range}(A^*) = \operatorname{range}(A^*A) = \operatorname{range}(A^\dagger A)$.

(c) $\ker(A) = \ker(A^*A) = \ker(A^\dagger A)$.

(d) $\ker(A^\dagger) = \ker(A^*) = \ker(AA^*) = \ker(AA^\dagger)$.

(e) $\operatorname{range}(A^\dagger A) \oplus \ker(A^\dagger A) = F^n$.

(f) $\operatorname{range}(AA^\dagger) \oplus \ker(AA^\dagger) = F^m$.

22. If $A = A_1 + A_2 + \cdots + A_k$, $A_i^* A_j = 0$, and $A_i A_j^* = 0$, for all $i, j = 1, \cdots, k$, $i \neq j$, then $A^\dagger = A_1^\dagger + A_2^\dagger + \cdots + A_k^\dagger$.
23. If A is an $m\times r$ matrix of rank r and B is an $r\times n$ matrix of rank r, then $(AB)^\dagger = B^\dagger A^\dagger$.
24. $(A^*A)^\dagger = A^\dagger(A^*)^\dagger$; $(AA^*)^\dagger = (A^*)^\dagger A^\dagger$.
25. [Gre66] Each one of the following conditions is necessary and sufficient for $(AB)^\dagger = B^\dagger A^\dagger$:

(a) $\operatorname{range}(BB^*A^*) \subseteq \operatorname{range}(A^*)$ and $\operatorname{range}(A^*AB) \subseteq \operatorname{range}(B)$.

(b) $A^\dagger ABB^*$ and $A^*ABB^\dagger$ are both Hermitian matrices.

(c) $A^\dagger ABB^*A^* = BB^*A^*$ and $BB^\dagger A^*AB = A^*AB$.

(d) $A^\dagger ABB^*A^*ABB^\dagger = BB^*A^*A$.

(e) $A^\dagger AB = B(AB)^\dagger AB$ and $BB^\dagger A^* = A^*AB(AB)^\dagger$.

26. $(A\otimes B)^\dagger = A^\dagger \otimes B^\dagger$, where $\otimes$ denotes the Kronecker product.
27. $A^\dagger = \lim_{\alpha\to 0} A^*(\alpha I + AA^*)^{-1} = \lim_{\alpha\to 0}(\alpha I + A^*A)^{-1}A^*$.
28. $A^\dagger = \sum_{j=1}^{\infty} A^*(I + AA^*)^{-j} = \sum_{j=1}^{\infty}(I + A^*A)^{-j}A^*$.
29. (Continuity of pseudo-inverse) Suppose that $A \in F^{m\times n}$ and $E \in F^{m\times n}$, where $F = \mathbb{C}$ or $\mathbb{R}$. Then $\lim_{E\to 0}(A+E)^\dagger = A^\dagger$ if and only if there is $\epsilon > 0$ such that $\operatorname{rank}(A+E) = \operatorname{rank}(A)$ when $\|E\|_2 \le \epsilon$.
30. Let $A \in \mathbb{C}^{m\times n}$ be of rank r where $0 < r < \min\{m,n\}$. Suppose that A can be partitioned as

$$A = \begin{bmatrix} A_{11} & A_{12}\\ A_{21} & A_{22}\end{bmatrix},$$

where $A_{11} \in \mathbb{C}^{r\times r}$ and $\operatorname{rank}(A_{11}) = r$. Then

$$A^\dagger = \begin{bmatrix} A_{11}^*XA_{11}^* & A_{11}^*XA_{21}^*\\ A_{12}^*XA_{11}^* & A_{12}^*XA_{21}^*\end{bmatrix},$$

where

$$X = (A_{11}A_{11}^* + A_{12}A_{12}^*)^{-1}A_{11}(A_{11}^*A_{11} + A_{21}^*A_{21})^{-1}.$$

Applications

- Handbook of Lin. Algebra (20 ✔ / 6 ✔ / 4 ✖) (Bernauer, H., Regensburger '23)

Bernauer, H., Regensburger. *How to Automatise Proofs of Operator Statements: Moore-Penrose Inverse; A Case Study*. In: Proceedings of CASC. 2023.

Applications

- Handbook of Lin. Algebra (20 ✔/ 6 ✔/ 4 ✖) (Bernauer, H., Regensburger '23)
  - yields ideals with $\leqslant 70$ generators in $\leqslant 18$ indeterminates
  - all proofs take $\sim$17 seconds altogether

Bernauer, H., Regensburger. *How to Automatise Proofs of Operator Statements: Moore-Penrose Inverse; A Case Study*. In: Proceedings of CASC. 2023.

Applications

- Handbook of Lin. Algebra (20 ✔/ 6 ✔/ 4 ✖) (Bernauer, H., Regensburger '23)
  - yields ideals with $\leqslant 70$ generators in $\leqslant 18$ indeterminates
  - all proofs take ~17 seconds altogether
- Recent results in operator theory (Bernauer, H., Regensburger '23)
  - *Reverse order law of the Moore-Penrose inverse* (Djordjević, Dinčić '09)
  - they: *We use [. . .] decompositions of Hilbert spaces*
  - we: purely algebraic proofs $\Rightarrow$ our proofs generalise results

Bernauer, H., Regensburger. *How to Automatise Proofs of Operator Statements: Moore-Penrose Inverse; A Case Study*. In: Proceedings of CASC. 2023.

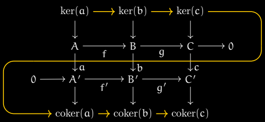
Applications

- Handbook of Lin. Algebra (20 ✔/ 6 ✔/ 4 ✖) (Bernauer, H., Regensburger '23)
  - yields ideals with $\leqslant 70$ generators in $\leqslant 18$ indeterminates
  - all proofs take ∼17 seconds altogether
- Recent results in operator theory (Bernauer, H., Regensburger '23)
  - *Reverse order law of the Moore-Penrose inverse* (Djordjević, Dinčić '09)
  - they: *We use [. . .] decompositions of Hilbert spaces*
  - we: purely algebraic proofs $\Rightarrow$ our proofs generalise results
- New results (Cvetković-Ilić, H., Hossein Poor, Milošević, Raab, Regensburger '21)
  - software used to find minimal assumptions

Bernauer, H., Regensburger. *How to Automatise Proofs of Operator Statements: Moore-Penrose Inverse; A Case Study*. In: Proceedings of CASC. 2023.
Cvetković-Ilić, H., Hossein Poor, Milošević, Raab, Regensburger. *Algebraic proof methods for identities of matrices and operators: improvements of Hartwig's triple reverse order law*. In: Applied Mathematics and Computation. 2021.

Applications

- Handbook of Lin. Algebra (20 ✔/ 6 ✔/ 4 ✘) (Bernauer, H., Regensburger '23)
  - yields ideals with $\leqslant 70$ generators in $\leqslant 18$ indeterminates
  - all proofs take ~17 seconds altogether
- Recent results in operator theory (Bernauer, H., Regensburger '23)
  - *Reverse order law of the Moore-Penrose inverse* (Djordjević, Dinčić '09)
  - they: *We use [...] decompositions of Hilbert spaces*
  - we: purely algebraic proofs $\Rightarrow$ our proofs generalise results
- New results (Cvetković-Ilić, H., Hossein Poor, Milošević, Raab, Regensburger '21)
  - software used to find minimal assumptions
- Diagram lemmas (Five lemma, Nine lemma, Snake lemma,...)



Bernauer, H., Regensburger. *How to Automatise Proofs of Operator Statements: Moore-Penrose Inverse; A Case Study*. In: Proceedings of CASC. 2023.

Cvetković-Ilić, H., Hossein Poor, Milošević, Raab, Regensburger. *Algebraic proof methods for identities of matrices and operators: improvements of Hartwig's triple reverse order law*. In: Applied Mathematics and Computation. 2021.

operator_gb

$$
\begin{aligned}
= \text{ nc Gröbner bases} + \text{certified ideal membership } &+ \text{ ideal arithmetic} \\
&+ \text{ heuristics} \\
&+ \text{ operator auxiliaries}
\end{aligned}
$$

operator_gb

= nc Gröbner bases + certified ideal membership + ideal arithmetic

Opal, Bergman, Magma, NCAlgebra (Mathematica), GAP, NCPoly (ApCoCoA)

Letterplace (Singular)

+ heuristics

+ operator auxiliaries

operator_gb

= nc Gröbner bases + certified ideal membership + ideal arithmetic

Opal, Bergman, Magma, NCAlgebra (Mathematica), GAP, NCPoly (ApCoCoA)

Letterplace (Singular)

+ heuristics

+ operator auxiliaries

Foundation: efficient noncommutative F4 algorithm

Requires: fast monomial comparisons + fast (sparse) linear algebra

operator_gb

= nc Gröbner bases + certified ideal membership + ideal arithmetic + heuristics + operator auxiliaries

nc Gröbner bases: Opal, Bergman, Magma, NCAlgebra (Mathematica), GAP, NCPoly (ApCoCoA)

certified ideal membership: Letterplace (Singular)

Foundation: efficient noncommutative F4 algorithm

Requires: fast monomial comparisons + fast (sparse) linear algebra

monomials = strings $\rightsquigarrow$ efficient multi-pattern string matching

dedicated (sparse) LA in C (via Cython) exploiting matrix structure

operator_gb

= nc Gröbner bases + certified ideal membership + ideal arithmetic + heuristics + operator auxiliaries

nc Gröbner bases	certified ideal membership
Opal, Bergman, Magma, NCAlgebra (Mathematica), GAP, NCPoly (ApCoCoA)	Letterplace (Singular)

Foundation: efficient noncommutative F4 algorithm

Requires: fast monomial comparisons + fast (sparse) linear algebra

monomials = strings
$\rightsquigarrow$
efficient multi-pattern string matching

dedicated (sparse) LA in C (via Cython) exploiting matrix structure

signature_gb = Noncommutative signature Gröbner bases

Signature Gröbner bases

Observation Lots of redundant operations in GB computations

Goal Detect these operations!

$f = \sum a_{i,j} \cdot f_i \cdot b_{i,j}$

$\sigma = \mathrm{lt}(\sum a_{i,j} \cdot \varepsilon_i \cdot b_{i,j})$

Idea Add "birth certificate" to polynomials $f \rightsquigarrow (f, \sigma)$

Signature Gröbner bases

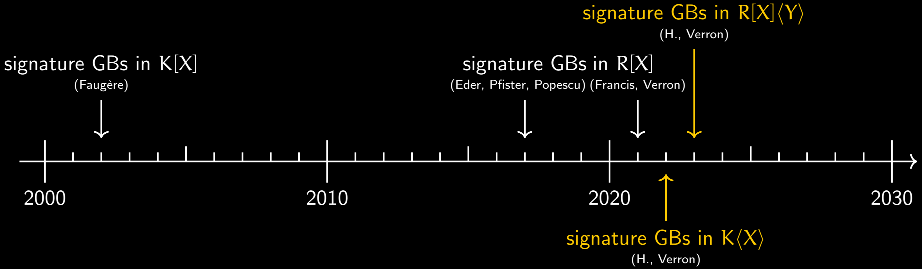
Observation Lots of redundant operations in GB computations

Goal Detect these operations!

Idea Add "birth certificate" to polynomials $f \rightsquigarrow (f, \sigma)$

$$f = \sum a_{i,j} \cdot f_i \cdot b_{i,j}$$

$$\sigma = \mathrm{lt}\left(\sum a_{i,j} \cdot \varepsilon_i \cdot b_{i,j}\right)$$



H., Verron. *Signature Gröbner bases, bases of syzygies and cofactor reconstruction in the free algebra*. In: Journal of Symbolic Computation. 2022.

H., Verron. *Signature Gröbner Bases in Free Algebras over Rings*. In: Proceedings of ISSAC. 2023.

Signature Gröbner bases

Observation Lots of redundant operations in GB computations

Goal Detect these operations!

Idea Add “birth certificate” to polynomials $f \rightsquigarrow (f, \sigma)$

$f = \sum a_{i,j} \cdot f_i \cdot b_{i,j}$

$\sigma = \operatorname{lt}(\sum a_{i,j} \cdot \varepsilon_i \cdot b_{i,j})$

In the noncommutative setting...

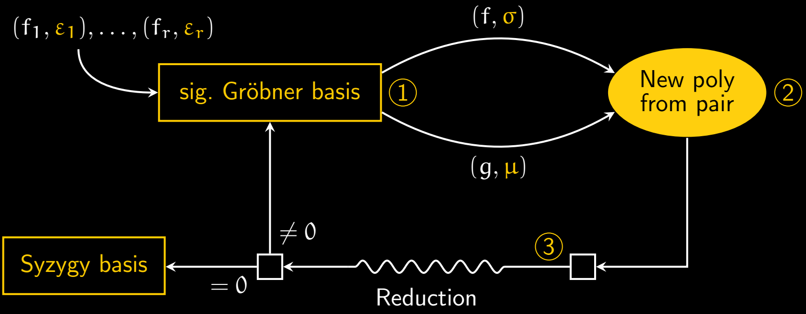
...many things are similar (basic definitions, algorithm blueprint)

...many things are very different (trivial syzygies, handling of S-polynomials, decoupling selection strategy from signature order)

H., Verron. *Signature Gröbner bases, bases of syzygies and cofactor reconstruction in the free algebra*. In: Journal of Symbolic Computation. 2022.

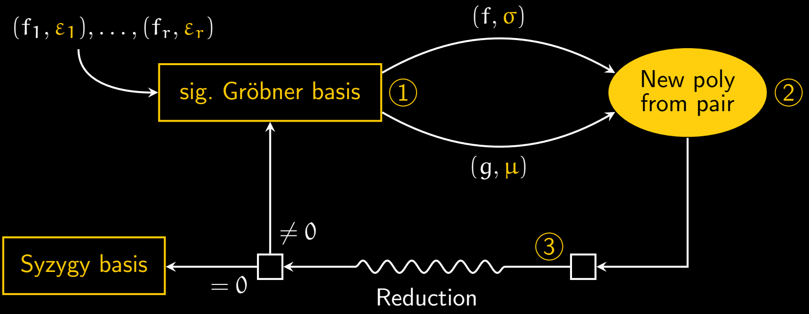
H., Verron. *Signature Gröbner Bases in Free Algebras over Rings*. In: Proceedings of ISSAC. 2023.

Sig-based Buchberger's algorithm



1. Selection: fair strategy *"Every S/G-poly is selected eventually"*
2. Construction: regular S/G-polynomials
3. Reduction: regular sig-reductions

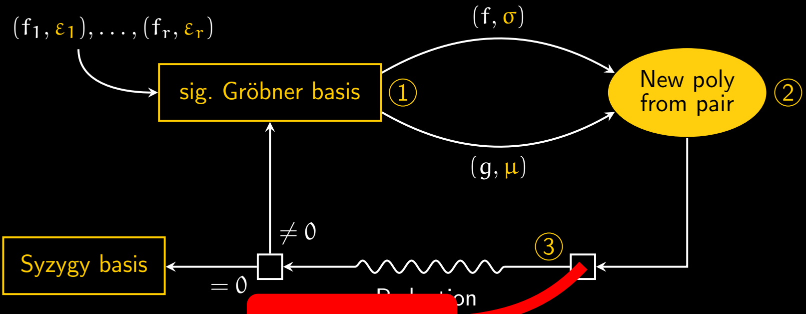
Sig-based Buchberger's algorithm



1. Selection: fair strategy *"Every S/G-poly is selected eventually"*
2. Construction: regular S/G-polynomials
3. Reduction: regular sig-reductions

Theorem (H., Verron '22, '23b)
This enumerates a (possibly infinite) sig. GB and syzygy basis

Sig-based Buchberger's algorithm



1. **Selection:** fair str… G-poly is selected eventually”
2. **Construction:** regular S/G-polynomials
3. **Reduction:** regular sig-reductions

Theorem (H., Verron '22, '23b)
This enumerates a (possibly infinite) sig. GB and syzygy basis

operator_gb **vs.** signature_gb

Hilbert's 24th problem

"The 24th problem in my Paris lecture was to be: Criteria of simplicity, or proof of the greatest simplicity of certain proofs.

- David Hilbert, ∼1900.

Hilbert's 24th problem

"The 24th problem in my Paris lecture was to be: Criteria of simplicity, or proof of the greatest simplicity of certain proofs.

- David Hilbert, ~1900.

operator statement ⇜⇝ proof

↕ ↕

ideal membership ⇜⇝ cofactor representation

$f \in (f_1, \ldots, f_r)$ $f = \sum_{i,j=1}^{N} a_{i,j} \cdot f_i \cdot b_{i,j}$

Hilbert's 24th problem

"The 24th problem in my Paris lecture was to be: Criteria of simplicity, or proof of the greatest simplicity of certain proofs.

- David Hilbert, ~1900.

operator statement ⇜⇝ proof

$\updownarrow$ $\qquad$ $\updownarrow$

ideal membership ⇜⇝ cofactor representation

$f \in (f_1, \ldots, f_r)$ $\qquad$ $f = \sum_{i,j=1}^{N} a_{i,j} \cdot f_i \cdot b_{i,j}$

Can we decide whether a cofactor representation exists?

Hilbert's 24th problem

"The 24th problem in my Paris lecture was to be: Criteria of simplicity, or proof of the greatest simplicity of certain proofs.

- David Hilbert, ~1900.

operator statement ⇜⇝ proof

↕ ↕

ideal membership ⇜⇝ cofactor representation

$f \in (f_1, \dots, f_r)$ $f = \sum_{i,j=1}^{N} a_{i,j} \cdot f_i \cdot b_{i,j}$

Can we decide whether a cofactor representation exists? – No.

Hilbert's 24th problem

"The 24th problem in my Paris lecture was to be: Criteria of simplicity, or proof of the greatest simplicity of certain proofs.

- David Hilbert, ~1900.

operator statement $\leftrightsquigarrow$ proof

$\updownarrow$ $\qquad$ $\updownarrow$

ideal membership $\leftrightsquigarrow$ cofactor representation

$f \in (f_1, \ldots, f_r)$ $\qquad$ $f = \sum_{i,j=1}^{N} a_{i,j} \cdot f_i \cdot b_{i,j}$

Can we decide whether a cofactor representation of length $\leqslant N$ exists? – Yes!

Hilbert's 24th problem

"The 24th problem in my Paris lecture was to be: Criteria of simplicity, or proof of the greatest simplicity of certain proofs.

- David Hilbert, ~1900.

operator statement ⇝ proof

↕ ↕

ideal membership ⇝ cofactor representation

$f \in (f_1, \ldots, f_r)$ $f = \sum_{i,j=1}^{N} a_{i,j} \cdot f_i \cdot b_{i,j}$

Can we decide whether a cofactor representation of length $\leqslant N$ exists? – Yes!

Theorem (H., Verron '23a)

In a (minimal) cofactor representation

$$\max_{i,j} \deg(a_{i,j} \cdot f_i \cdot b_{i,j}) \leqslant \text{poly}(N, \deg(f), \deg(f_i)).$$

H., Verron. *Short proofs of ideal membership*. preprint. 2023.

Short proofs in practice

Theorem (Djordjević, Dinčić '09)

A, B matrices such that AB exists.

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

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

Short proofs in practice

Theorem (Djordjević, Dinčić '09)

A, B matrices such that AB exists.

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

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

Classical Gröbner bases: 1203 terms (~16 pages)

Short proofs in practice

Theorem (Djordjević, Dinčić '09)

A, B matrices such that AB exists.

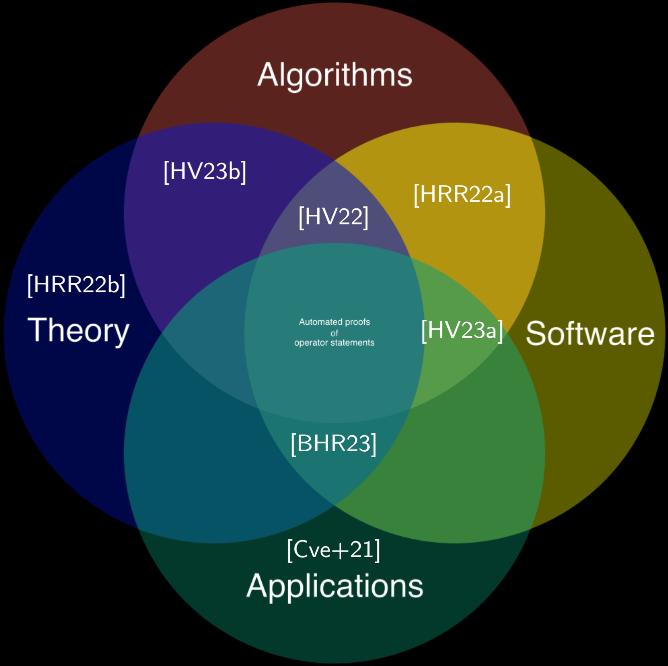
$$B^\dagger(ABB^\dagger)^\dagger = (A^\dagger AB)^\dagger A^\dagger = B^\dagger A^\dagger \quad \Rightarrow \quad (AB)^\dagger = B^\dagger A^\dagger$$

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

Classical Gröbner bases: 1203 terms (~16 pages)

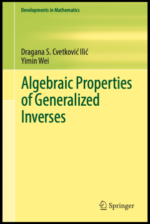
New approach:

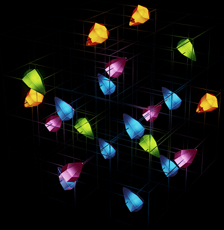
$$\begin{aligned}(ab)^\dagger - b^\dagger a^\dagger = \; & f_{21} - f_{10} + b^\dagger f_{14} - f_{12}(ab)^\dagger - b^\dagger(abb^\dagger)^\dagger f_{11} + b^\dagger(abb^\dagger)^\dagger f_{15} \\ & + (a^\dagger ab)^\dagger a^\dagger f_9(ab)^\dagger - b^* f_{23}((ab)^\dagger)^*(ab)^\dagger - f_{21}ab(ab)^\dagger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


Algorithms
[HV23b]
[HRR22a]
[HV22]
[HRR22b]
Theory
Automated proofs
of
operator statements
[HV23a]
Software
[BHR23]
[Cve+21]
Applications

Outlook

- Producing proof certificates
- More advanced computational techniques
  - Boolean abstraction (DPLL(T), CDCL(T))
  - Congruence closure
  - Unification
  - ⋮
- Further applications
  - Generalised inverses
  - Abelian categories
  - Group theory





- Constructing counterexamples
- Investigating structure of Gröbner bases