Automated proofs of operator statements

Clemens Hofstadler · Institute of Mathematics · University of Kassel
joint work with Clemens G. Raab and Georg Regensburger

Tagung Fachgruppe Computeralgebra
Hannover, Germany, June 1, 2023

UNIKASSEL VERSITÄT

FWF Der Wissenschaftsfonds.

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 $(\mathbf{xy}^*)^\dagger = \dfrac{\mathbf{yx}^*}{\|\mathbf{x}\|^2\|\mathbf{y}\|^2}$.
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) $\operatorname{range}(A^\dagger) = \operatorname{range}(\mathrm{A}^*) = \operatorname{range}(A^*A) = \operatorname{range}(A^\dagger A)$.
   (c) $\ker(A) = \ker(\mathrm{A}^*\mathrm{A}) = \ker(A^\dagger A)$.
   (d) $\ker(A^\dagger) = \ker(\mathrm{A}^*) = \ker(AA^*) = \ker(AA^\dagger)$.
   (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 $\operatorname{rank}(A+E) = \operatorname{rank}(\mathrm{A})$ 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
$$X = (A_{11}A_{11}^* + A_{12}A_{12}^*)^{-1}A_{11}(A_{11}^*A_{11} + A_{21}^*A_{21})^{-1}.$$

Theory

- Model linear operators by noncomm. polynomials
- Correctness of first-order operator statements
  $\Longleftrightarrow$
  nc ideal membership
- Approach is complete
  $\rightarrow$ Semi-decision procedure

Theory

- Model linear operators by noncomm. polynomials
- Correctness of first-order operator statements
  $\Longleftrightarrow$
  nc ideal membership
- Approach is complete
  $\rightarrow$ Semi-decision procedure

Software

- SageMath package `operator_gb`*
- Noncomm. Gröbner bases
- Certified nc ideal membership
- Noncomm. ideal arithmetic
- Dedicated methods for proving operator statements

* available at https://github.com/ClemensHofstadler/operator_gb

Theory

- Model linear operators by noncomm. polynomials
- Correctness of first-order operator statements
  $\Longleftrightarrow$
  nc ideal membership
- Approach is complete
  $\rightarrow$ Semi-decision procedure

Software

- SageMath package `operator_gb`*
- Noncomm. Gröbner bases
- Certified nc ideal membership
- Noncomm. ideal arithmetic
- Dedicated methods for proving operator statements

* available at https://github.com/ClemensHofstadler/operator_gb

Automated proofs of operator statements

Noncommutative polynomials

Noncommutative polynomials $=$ elements in free algebra $\mathbb{Z}\langle X \rangle$

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

Noncommutative polynomials

Noncommutative polynomials $=$ elements in free algebra $\mathbb{Z}\langle X \rangle$

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

finite words over X

Noncommutative polynomials

Noncommutative polynomials $=$ elements in free algebra $\mathbb{Z}\langle X\rangle$

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

finite words over X

Multiplication $=$ Concatenation of words

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

Noncommutative polynomials

Noncommutative polynomials $=$ elements in free algebra $\mathbb{Z}\langle X\rangle$

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

$x_{i,1} \dots x_{i,k_i}$: finite words over X

Multiplication $=$ Concatenation of words

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

Two-sided ideals For $f_1, \dots, f_r \in \mathbb{Z}\langle X\rangle$

$$(f_1, \dots, f_r) = \left\{ \sum_i \sum_j a_{i,j} f_i b_{i,j} \;\middle|\; a_{i,j}, b_{i,j} \in \mathbb{Z}\langle X\rangle \right\}$$

Noncommutative polynomials

Noncommutative polynomials $=$ elements in free algebra $\mathbb{Z}\langle X\rangle$

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

$x_{i,1} \dots x_{i,k_i}$: finite words over X

Multiplication $=$ Concatenation of words

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

Two-sided ideals For $f_1, \dots, f_r \in \mathbb{Z}\langle X\rangle$

$$(f_1, \dots, f_r) = \left\{ \sum_i \sum_j a_{i,j} f_i b_{i,j} \;\middle|\; a_{i,j}, b_{i,j} \in \mathbb{Z}\langle X\rangle \right\}$$

Fact Ideal membership problem $f \stackrel{?}{\in} (f_1, \dots, f_r)$ 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 $^*, .^T, \|\cdot\|, \otimes, \ldots$

- $0, a, b, c, \ldots$
- $s + t,\ s \cdot t,\ f(t_1, \ldots, t_n)$

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

Operator statements

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

- $0, a, b, c, \ldots$
- $s + t,\ s \cdot t,\ f(t_1, \ldots, t_n)$

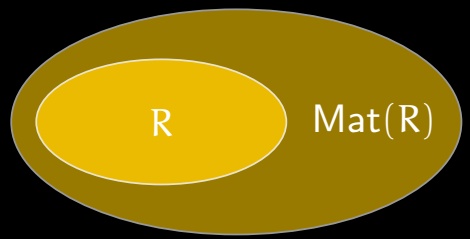
Linearity = abelian (partial) addition + assoc. (partial) mult. + dist.

Operator statements

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

- $0, a, b, c, \ldots$
- $s + t,\ s \cdot t,\ f(t_1, \ldots, t_n)$

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

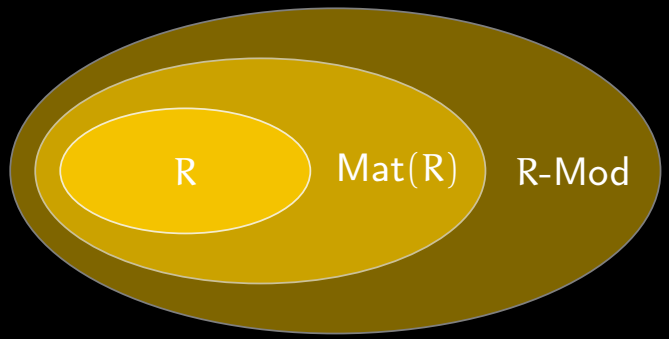


Operator statements

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

- $0, a, b, c, \ldots$
- $s + t,\ s \cdot t,\ f(t_1, \ldots, t_n)$

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



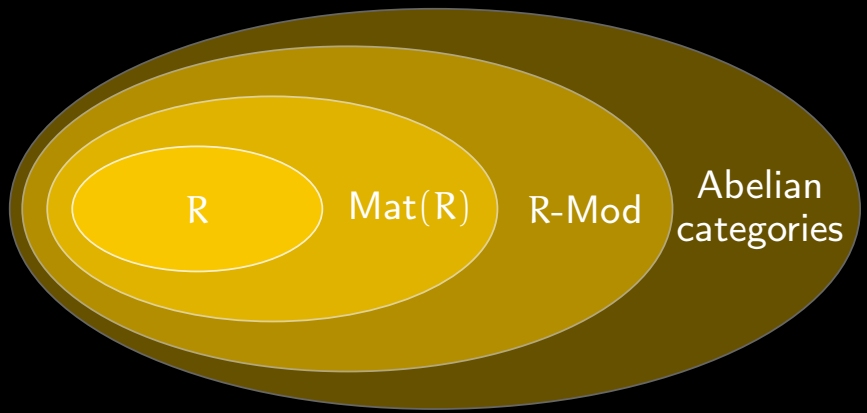
Operator statements

Operators

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

- $0, a, b, c, \ldots$
- $s + t,\ s \cdot t,\ f(t_1, \ldots, t_n)$

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



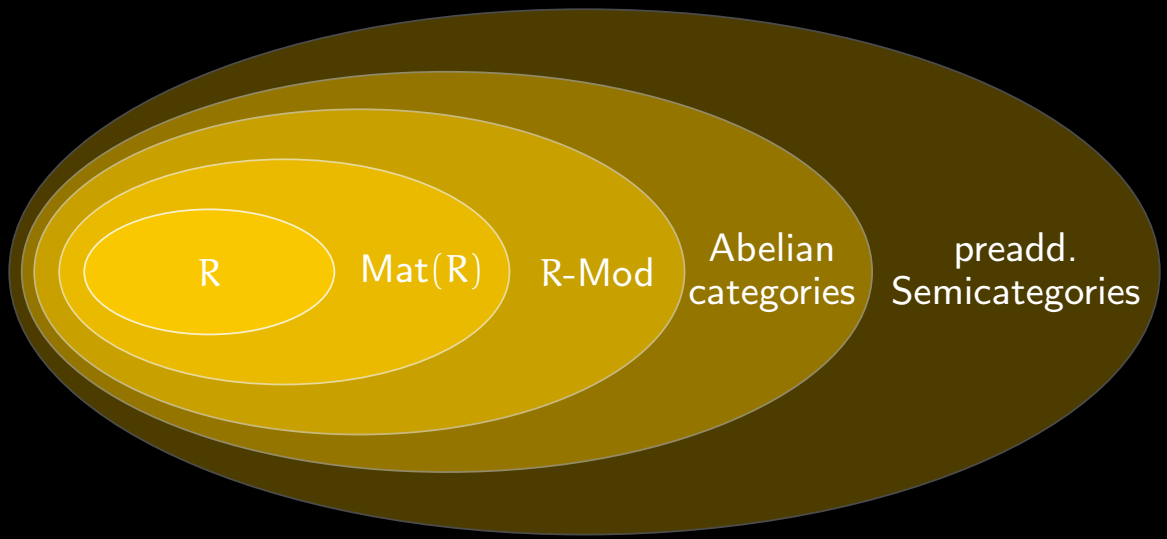
Operator statements

Operators

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

- $0, a, b, c, \ldots$
- $s + t,\ s \cdot t,\ f(t_1, \ldots, t_n)$

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



Operator statements

Operators

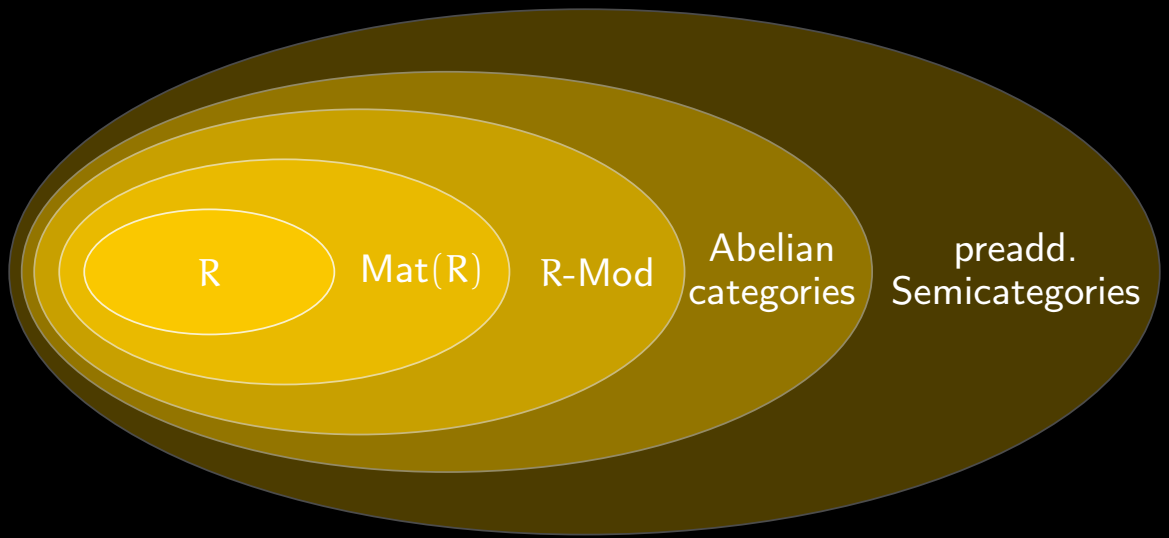
$^*, .^T, \|\cdot\|, \otimes, \ldots$

- $0, a, b, c, \ldots$
- $s + t,\ s \cdot t,\ f(t_1, \ldots, t_n)$

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

Operator statements

$s = t, \quad \neg\, \varphi, \quad (\varphi \wedge \psi), \quad (\varphi \vee \psi), \quad (\varphi \Rightarrow \psi), \quad \exists x : \varphi, \quad \forall x : \varphi$



Operator statements

Operators

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

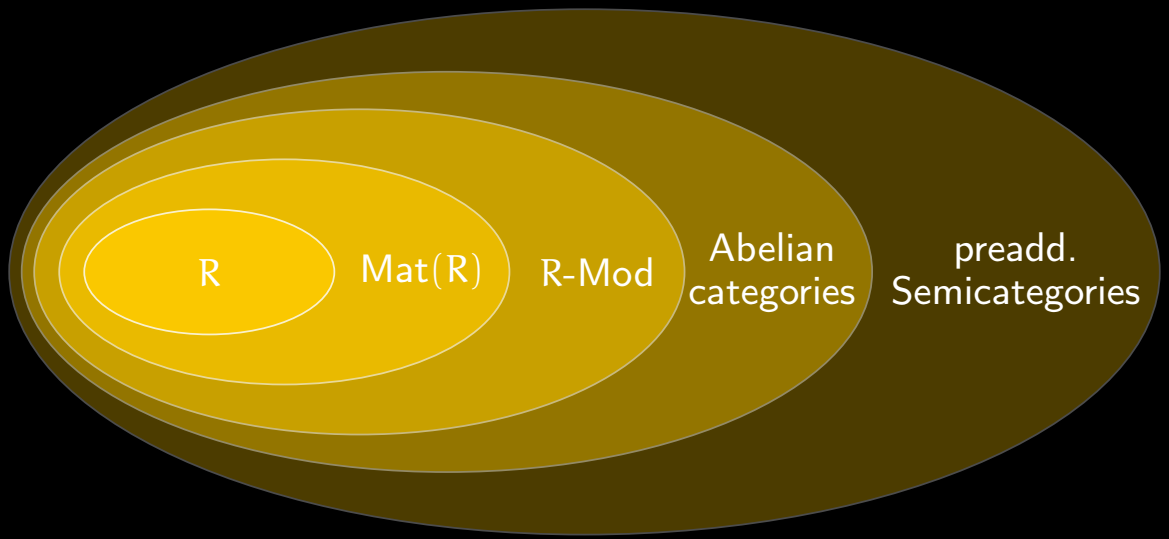
- $0, a, b, c, \ldots$
- $s + t,\ s \cdot t,\ f(t_1, \ldots, t_n)$

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

- $0, a, b, c, \ldots$
- $s + t,\ s \cdot t,\ f(t_1, \ldots, t_n)$

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

Operator statements

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

- $0, a, b, c, \ldots$
- $s + t,\ s \cdot t, f(t_1, \ldots, t_n)$

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: (efficient) semi-decision procedure
  $\rightarrow$ Can be obtained using computer algebra

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

Quasi-identities

- Classical case of quasi-identities well studied (Helton, Stankus, Wavrik '98, Schmitz, Levandovskyy '20, Raab, Regensburger, Hossein Poor '21)

$$\forall\, \mathbf{X} : \bigwedge_{j=1}^{m} A_j = B_j \Rightarrow P = Q$$

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

Quasi-identities

- Classical case of quasi-identities well studied (Helton, Stankus, Wavrik '98, Schmitz, Levandovskyy '20, Raab, Regensburger, Hossein Poor '21)

$$\forall\, \mathbf{X} : \bigwedge_{j=1}^{m} A_j = B_j \;\Rightarrow\; P = Q$$

Strategy

- Interpret each operator as polynomial in $\mathbb{Z}\langle \mathbf{X} \rangle$ and reformulate each identity $L = R$ as polynomial $L - R$

  e.g., $AB = BA \quad \rightsquigarrow \quad ab - ba \in \mathbb{Z}\langle a, b \rangle$

- "Being a consequence" ($\Rightarrow$) translates into ideal membership

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

Quasi-identities

- Classical case of quasi-identities well studied (Helton, Stankus, Wavrik '98, Schmitz, Levandovskyy '20, Raab, Regensburger, Hossein Poor '21)

$$\forall\, \mathbf{X} : \bigwedge_{j=1}^{m} A_j = B_j \;\Rightarrow\; P = Q$$

Strategy

- Interpret each operator as polynomial in $\mathbb{Z}\langle \mathbf{X} \rangle$ and reformulate each identity $L = R$ as polynomial $L - R$

  e.g., $AB = BA \quad \rightsquigarrow \quad ab - ba \in \mathbb{Z}\langle a, b \rangle$

- "Being a consequence" ($\Rightarrow$) translates into ideal membership

Theorem

$$\forall \mathbf{X} : \bigwedge_{j=1}^{m} A_j = B_j \Rightarrow P = Q \qquad \text{iff} \qquad p - q \in (a_1 - b_1, \ldots, a_m - b_m)$$

"The Moore-Penrose inverse is unique"

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

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

Claim If B and C satisfy these identities, then $\mathrm{B} = \mathrm{C}$

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

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

Claim If B and C satisfy these identities, then $\mathrm{B} = \mathrm{C}$

Proof Using our software package `operator_gb`...

"The Moore-Penrose inverse is unique"

Recall: B is Moore-Penrose inverse of A if

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

Claim If B and C satisfy these identities, then $\mathrm{B} = \mathrm{C}$

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

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

Claim If B and C satisfy these identities, then $\mathrm{B} = \mathrm{C}$

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

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

Claim If B and C satisfy these identities, then $\mathrm{B} = \mathrm{C}$

Proof Using our software package `operator_gb`...

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

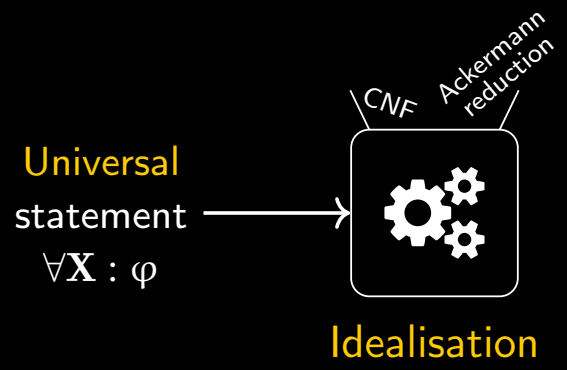
```
b - c = (-c + c*a*c) + b*c_adj*(-a_adj + a_adj*b_adj*a_adj)
        - b*a*c*(-a*b + b_adj*a_adj) - b*(-a + a*c*a)*b
        + b*(-a*c + c_adj*a_adj) - b*(-a*c + c_adj*a_adj)*b_adj*a_adj
        - (-b + b*a*b) + (-c*a + a_adj*c_adj)*b*a*c
        - (-a_adj + a_adj*c_adj*a_adj)*b_adj*c + c*(-a + a*b*a)*c
        - (-b*a + a_adj*b_adj)*c + a_adj*c_adj*(-b*a + a_adj*b_adj)*c
```

- Software produces cofactor representation
  (= certificate for ideal membership)
- Cofactor representation is algebraic proof requiring only linearity
  $\Rightarrow$ Statement is proven in all settings where linearity holds

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

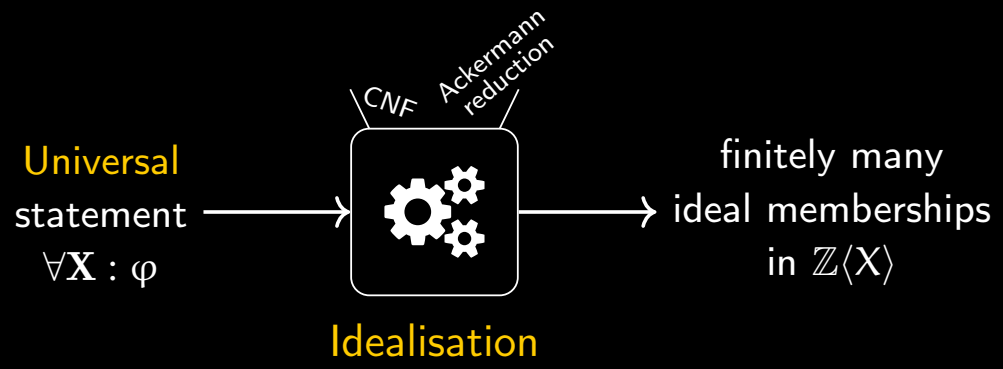
Universal statements



Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

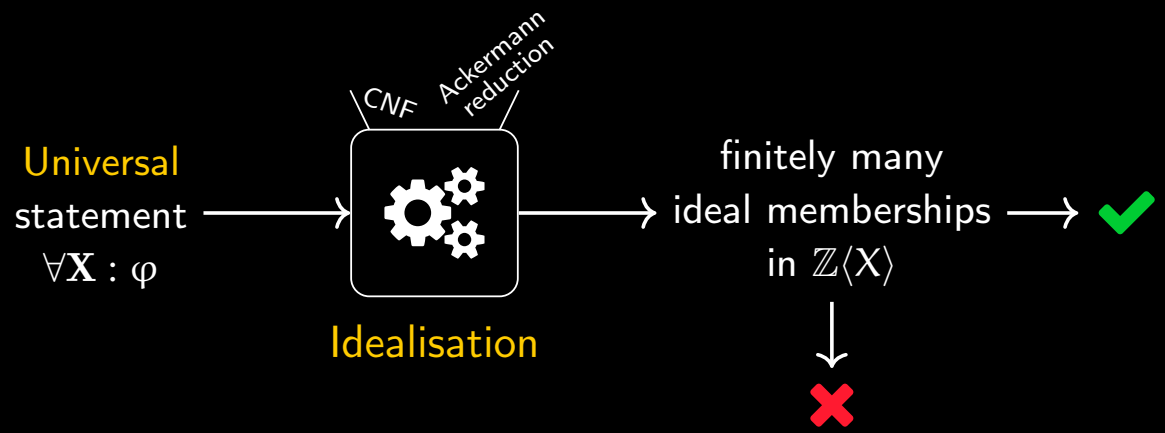
Universal statements



Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

Universal statements

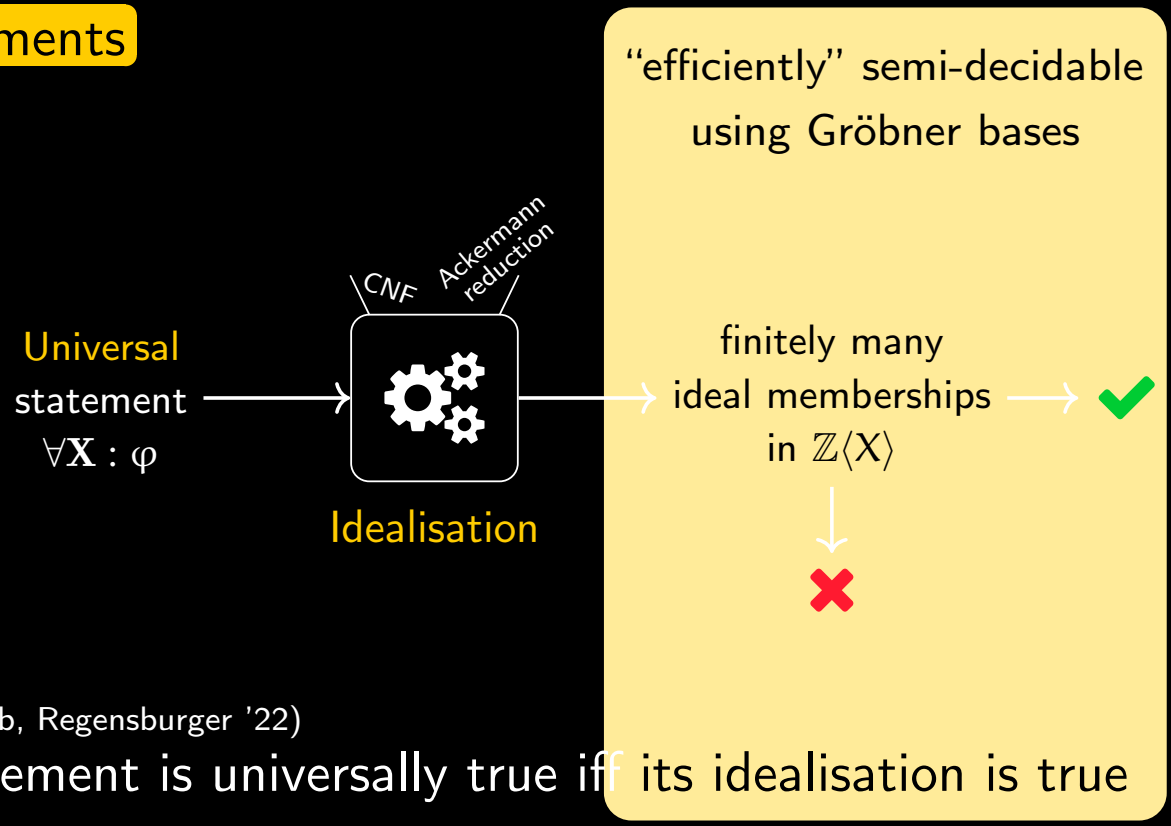


Theorem (H., Raab, Regensburger '22)
A universal statement is universally true iff its idealisation is true

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

Universal statements



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

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 $(\mathbf{xy}^*)^\dagger = \dfrac{\mathbf{yx}^*}{\|\mathbf{x}\|^2\|\mathbf{y}\|^2}$.
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) $\operatorname{range}(A^\dagger) = \operatorname{range}(\mathrm{A}^*) = \operatorname{range}(A^*A) = \operatorname{range}(A^\dagger A)$.
   (c) $\ker(A) = \ker(\mathrm{A}^*\mathrm{A}) = \ker(A^\dagger A)$.
   (d) $\ker(A^\dagger) = \ker(\mathrm{A}^*) = \ker(AA^*) = \ker(AA^\dagger)$.
   (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 $\operatorname{rank}(A+E) = \operatorname{rank}(\mathrm{A})$ 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 $\exists X : (PA^*A = A \wedge AA^*Q = A) \Rightarrow \mathsf{pinv}(A, X)$

"Every matrix has a Moore-Penrose inverse"

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

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

Strategy

1. Derive explicit expression for X
2. Reformulate statement as a universal statement
3. Prove by verifying ideal membership

"Every matrix has a Moore-Penrose inverse"

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

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

Strategy

1. Derive explicit expression for X
2. Reformulate statement as a universal statement
3. Prove by verifying ideal membership

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

Strategy

1. Derive explicit expression for X
2. Reformulate statement as a universal statement
3. Prove by verifying ideal membership

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

Strategy

1. Derive explicit expression for X
2. Reformulate statement as a universal statement
3. Prove by verifying ideal membership

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

Strategy

1. Derive explicit expression for X
2. Reformulate statement as a universal statement
3. Prove by verifying ideal membership

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 $\exists X : (PA^*A = A \wedge AA^*Q = A) \Rightarrow \text{pinv}(A, X)$

Strategy

1. Derive explicit expression for X
2. Reformulate statement as a universal statement
3. Prove by verifying ideal membership

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 expression.

Question Was this just luck?

Existential statements

In the previous example, we found a suitable expression.

Question Was this just luck? – No!

Existential statements

In the previous example, we found a suitable expression.

Question Was this just luck? – No!

Reason Herbrand's theorem (Herbrand '30)

An existential statement is universally true if and only if explicit expressions exist and can be constructed as polynomial expressions in terms of the basic operators appearing in the statement.

Existential statements

In the previous example, we found a suitable expression.

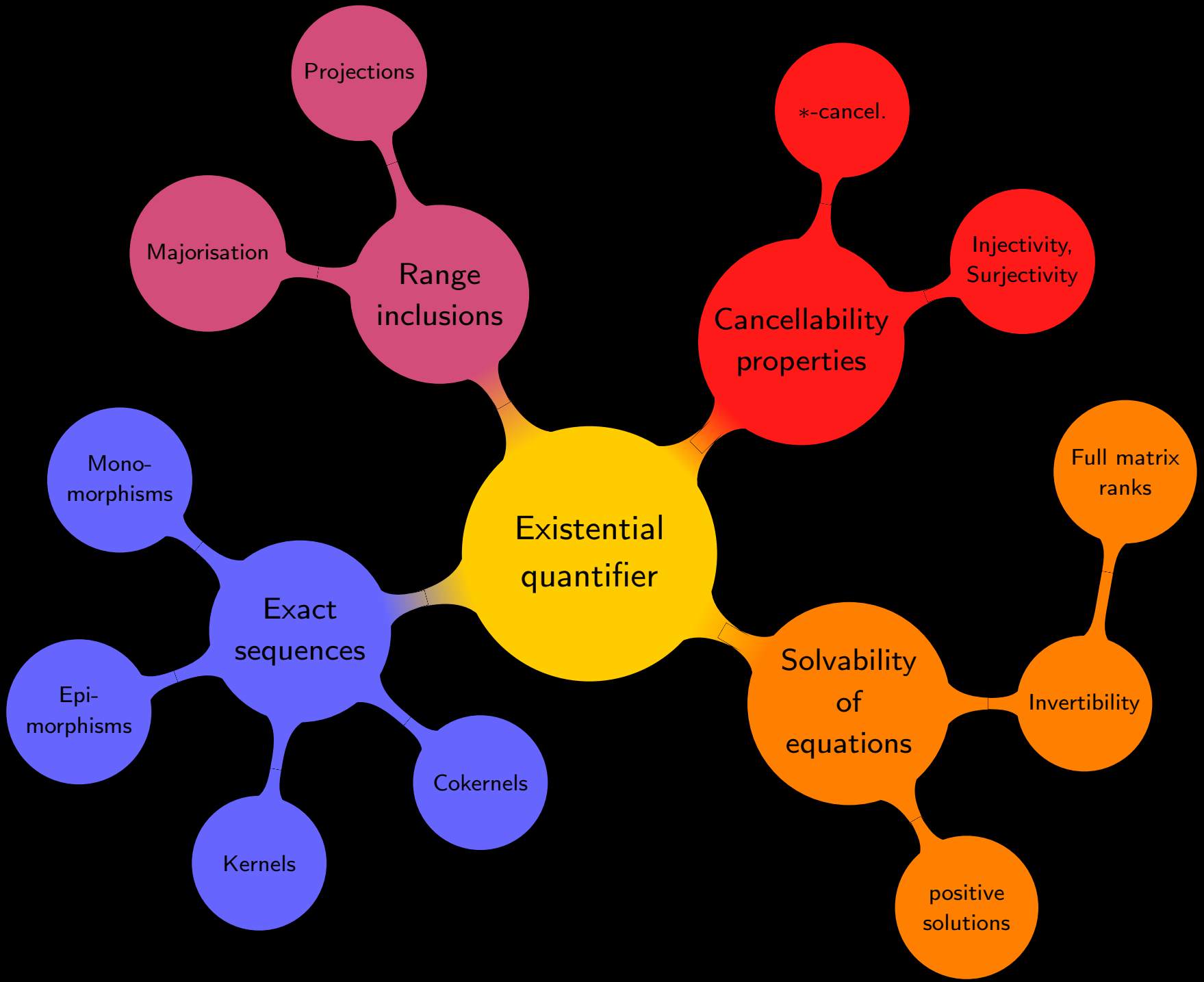
Question Was this just luck? – No!

Reason Herbrand's theorem (Herbrand '30)

An existential statement is universally true if and only if explicit expressions exist and can be constructed as polynomial expressions in terms of the basic operators appearing in the statement.

- Enumerating all possible expressions is hopeless
- Requires good heuristics $\longrightarrow$ provided by computer algebra
- Several heuristics implemented in `operator_gb`
  (ansatz, variable elimination, ideal/subalgebra intersections,. . .)

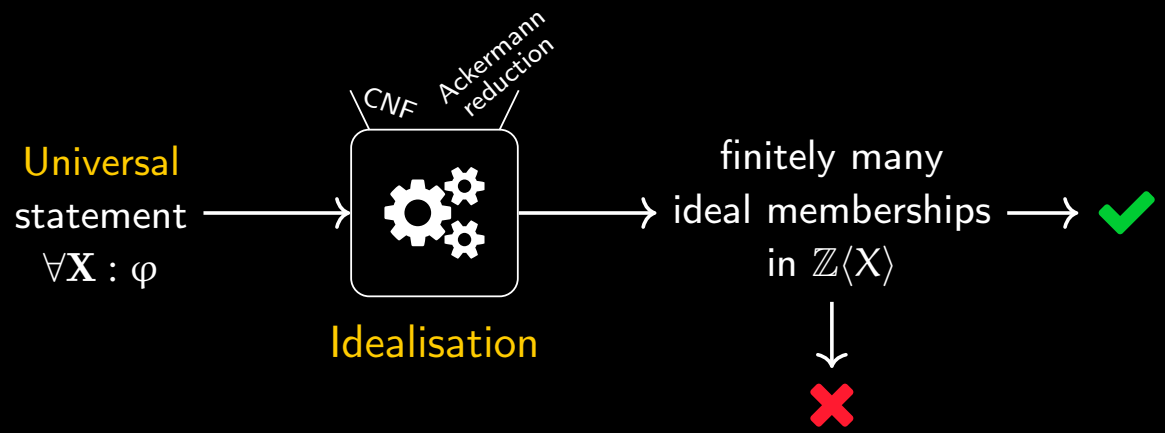

Existential
quantifier


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

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

Universal statements

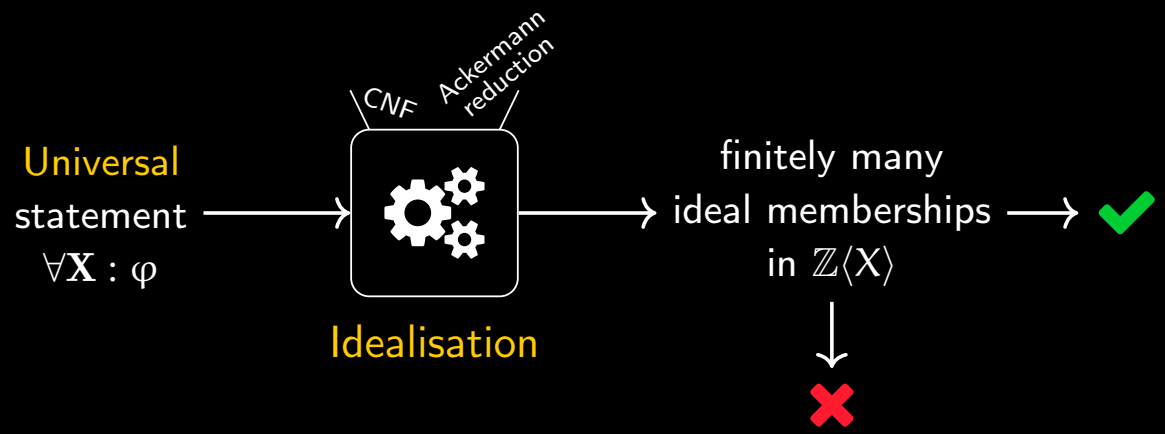


Theorem (H., Raab, Regensburger '22)
A universal statement is universally true iff its idealisation is true

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

Universal statements



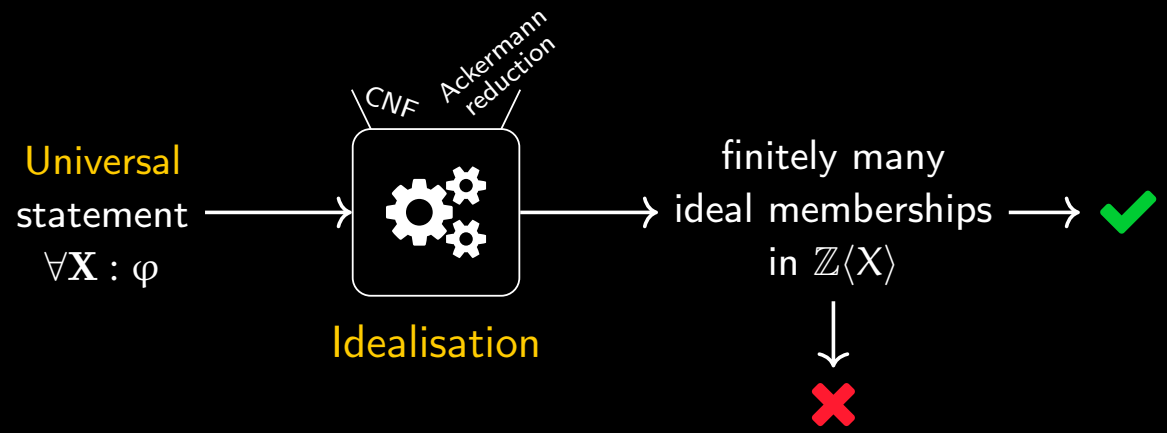
Theorem (H., Raab, Regensburger '22)
A universal statement is universally true iff its idealisation is true

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

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

General operator statements



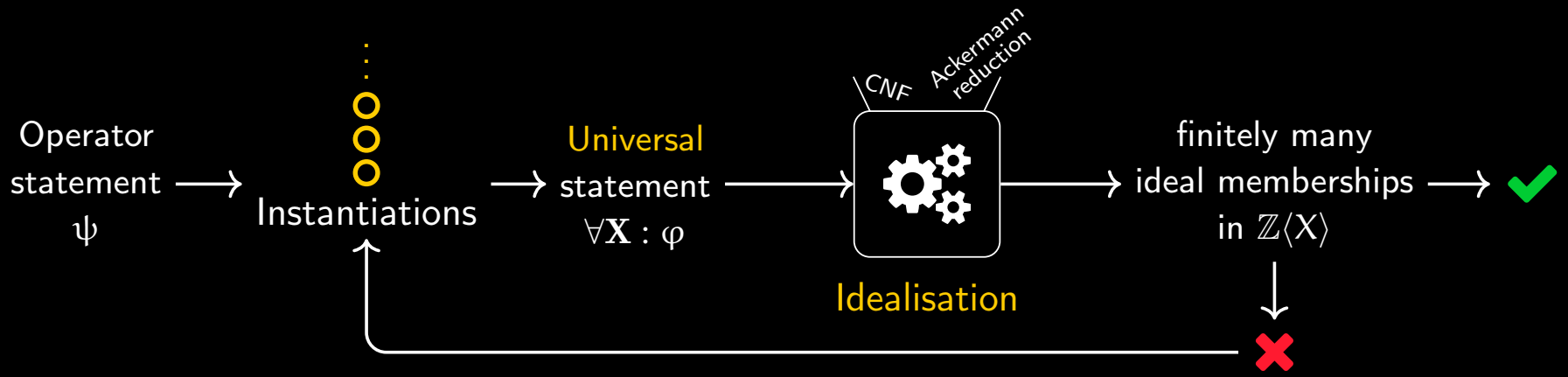
Theorem (H., Raab, Regensburger '22)
A universal statement is universally true iff its idealisation is true

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

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

General operator statements



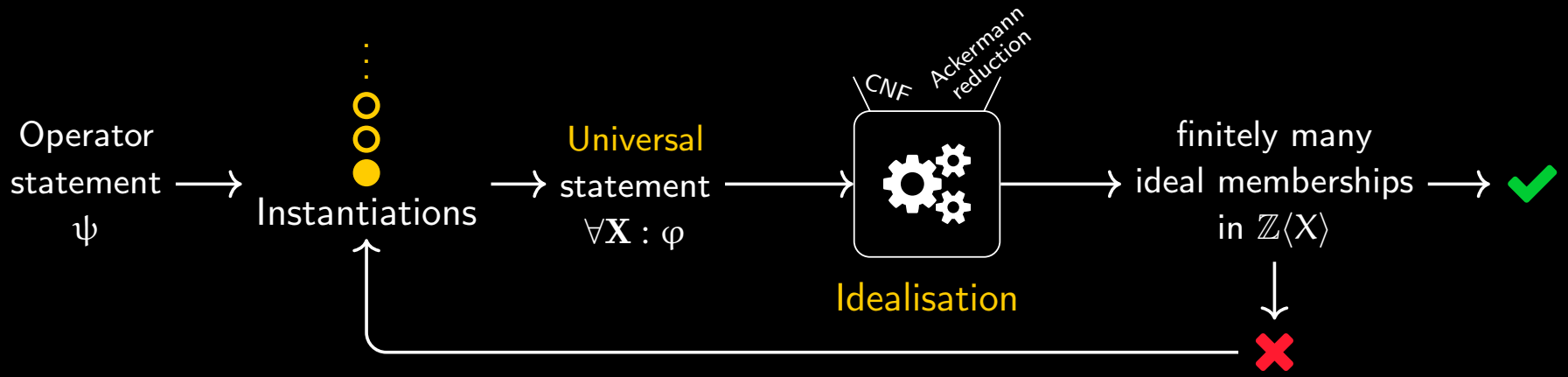
Theorem (H., Raab, Regensburger '22)
A universal statement is universally true iff its idealisation is true

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

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

General operator statements



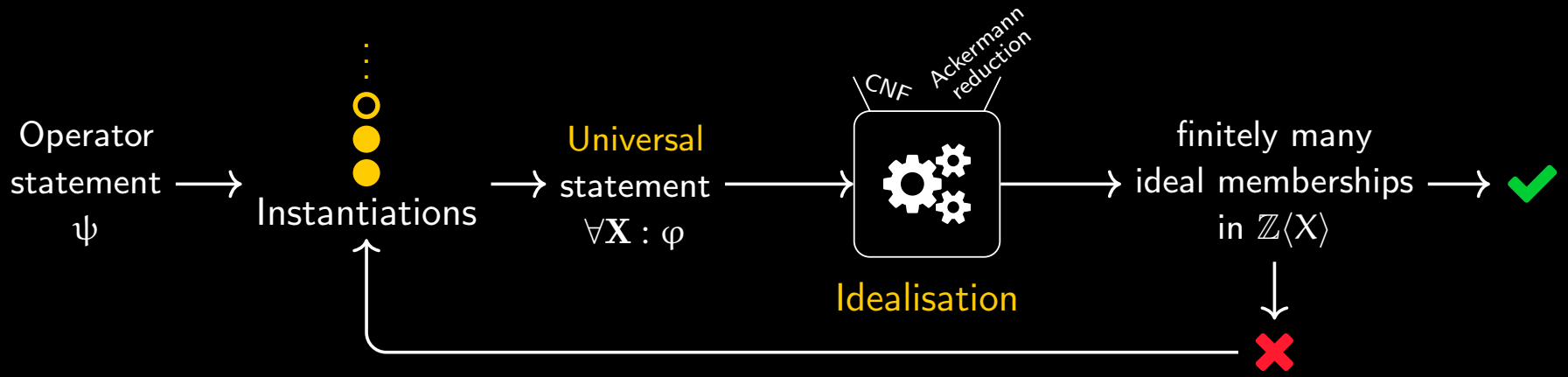
Theorem (H., Raab, Regensburger '22)
A universal statement is universally true iff its idealisation is true

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

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

General operator statements



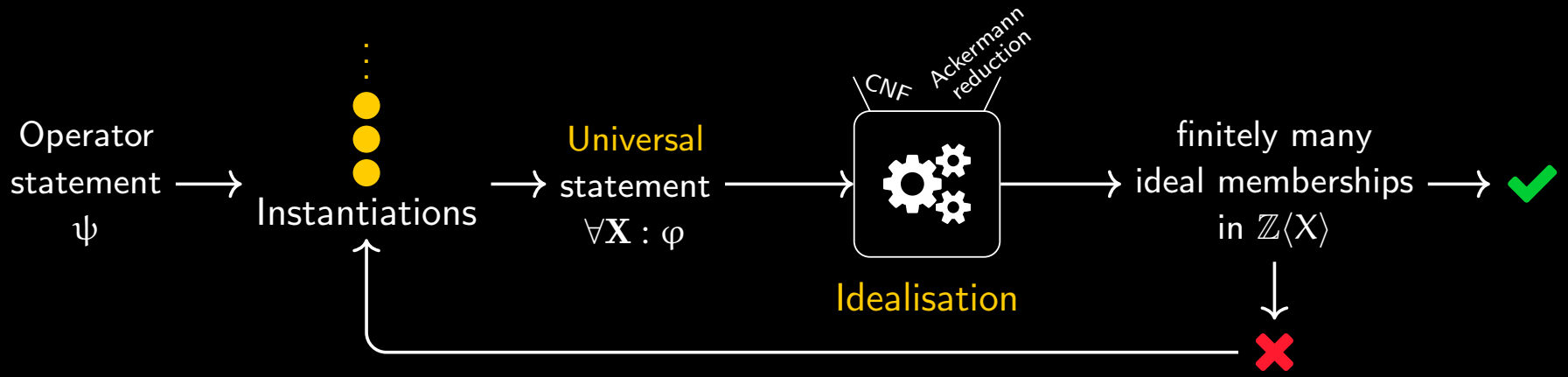
Theorem (H., Raab, Regensburger '22)
A universal statement is universally true iff its idealisation is true

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

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

General operator statements



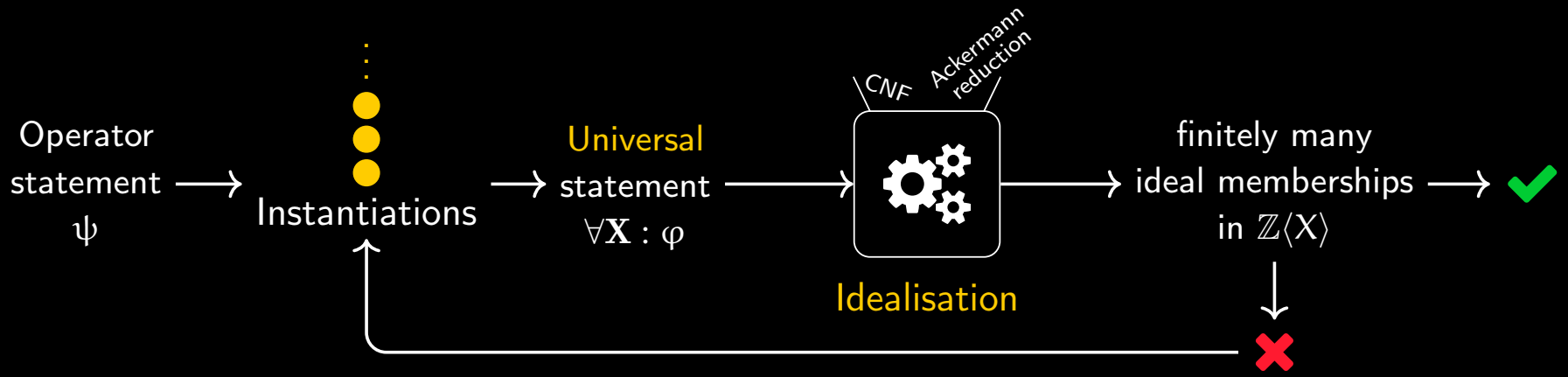
Theorem (H., Raab, Regensburger '22)
A universal statement is universally true iff its idealisation is true

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

Determining universal truth

Idea Translate universal truth of formula into polynomial predicate

General operator statements



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

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

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

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

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 $(\mathbf{xy}^*)^\dagger = \dfrac{\mathbf{yx}^*}{\|\mathbf{x}\|^2\|\mathbf{y}\|^2}$.
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) $(\mathrm{diag}(\beta_1, \beta_2, \cdots, \beta_n))^\dagger = \mathrm{diag}(\beta_1^\dagger, \beta_2^\dagger, \cdots, \beta_n^\dagger)$.
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) $\mathrm{rank}(A) = \mathrm{rank}(A^\dagger) = \mathrm{rank}(AA^\dagger) = \mathrm{rank}(A^\dagger A)$.
   (c) $\mathrm{rank}(I_n - A^\dagger A) = n - \mathrm{rank}(A)$.
   (d) $\mathrm{rank}(I_m - AA^\dagger) = m - \mathrm{rank}(A)$.
20. $AA^\dagger = \mathrm{Proj}_{\mathrm{range}(A)}$; $A^\dagger A = \mathrm{Proj}_{\mathrm{range}(A^\dagger)}$.
21. Suppose that $A \in F^{m\times n}$, where $F = \mathbb{C}$ or $\mathbb{R}$. Then
   (a) $\mathrm{range}(A) = \mathrm{range}(AA^*) = \mathrm{range}(AA^\dagger)$.
   (b) $\mathrm{range}(A^\dagger) = \mathrm{range}(A^*) = \mathrm{range}(A^*A) = \mathrm{range}(A^\dagger A)$.
   (c) $\ker(A) = \ker(A^*A) = \ker(A^\dagger A)$.
   (d) $\ker(A^\dagger) = \ker(A^*) = \ker(AA^*) = \ker(AA^\dagger)$.
   (e) $\mathrm{range}(A^\dagger A) \oplus \ker(A^\dagger A) = F^n$.
   (f) $\mathrm{range}(AA^\dagger) \oplus \ker(AA^\dagger) = F^m$.
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) $\mathrm{range}(BB^*A^*) \subseteq \mathrm{range}(A^*)$ and $\mathrm{range}(A^*AB) \subseteq \mathrm{range}(B)$.
   (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 $\mathrm{rank}(A+E) = \mathrm{rank}(A)$ 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 $\mathrm{rank}(A_{11}) = r$. 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 Linear Algebra (20 ✔ / 6 ✔ / 4 ✖)

Applications

- Handbook of Linear Algebra (20 ✔ / 6 ✔ / 4 ✖)
  - yields ideals with $\leqslant 70$ generators in $\leqslant 18$ indeterminates
  - cofactor representations consist of $\leqslant 226$ terms
  - all proofs take $\sim$15 seconds altogether

Applications

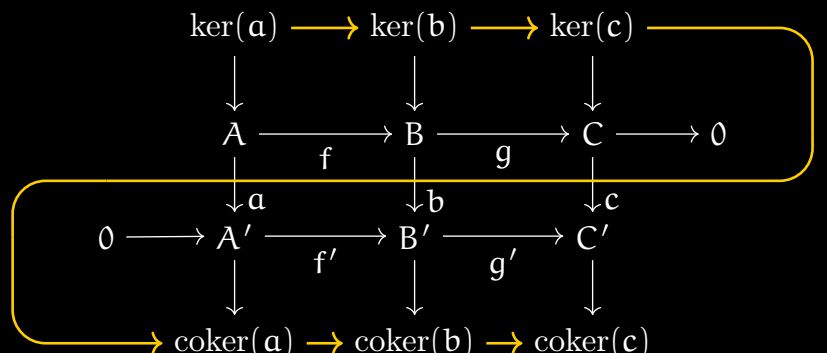
- Handbook of Linear Algebra (20 ✔ / 6 ✔ / 4 ✖)
  - yields ideals with $\leqslant 70$ generators in $\leqslant 18$ indeterminates
  - cofactor representations consist of $\leqslant 226$ terms
  - all proofs take ~15 seconds altogether
- Recent results in operator theory
  - *Reverse order law of the Moore-Penrose inverse* (Djordjević, Dinčić '09)
  - they: *We use [. . .] decompositions of Hilbert spaces*
  - we: purely algebraic proofs $\Rightarrow$ our proofs generalise results

Applications

- Handbook of Linear Algebra (20 ✔ / 6 ✔ / 4 ✖)
  - yields ideals with $\leqslant 70$ generators in $\leqslant 18$ indeterminates
  - cofactor representations consist of $\leqslant 226$ terms
  - all proofs take ~15 seconds altogether
- Recent results in operator theory
  - *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

Applications

- Handbook of Linear Algebra (20 ✔ / 6 ✔ / 4 ✖)
  - yields ideals with $\leqslant 70$ generators in $\leqslant 18$ indeterminates
  - cofactor representations consist of $\leqslant 226$ terms
  - all proofs take ~15 seconds altogether
- Recent results in operator theory
  - *Reverse order law of the Moore-Penrose inverse* (Djordjević, Dinčić '09)
  - they: *We use [. . .] decompositions of Hilbert spaces*
  - we: purely algebraic proofs ⇒ 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,. . .)



operator_gb

$=$ nc Gröbner bases $+$ certified ideal membership $+$ ideal arithmetic

$+$ heuristics

$+$ operator auxiliaries

operator_gb

= nc Gröbner bases + certified ideal membership + ideal arithmetic

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

+ heuristics

+ operator auxiliaries

operator_gb

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

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

Letterplace (Singular)

operator_gb

= nc Gröbner bases (Opal, Bergman, Magma, NCAlgebra (Mathematica), GAP, NCPoly (ApCoCoA)) + certified ideal membership (Letterplace (Singular)) + ideal arithmetic + heuristics + operator auxiliaries

Foundation: efficient noncommutative F4 algorithm

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

Realised via:

- Monomials are represented by (encoded) strings $\rightsquigarrow$ exploit efficient multi-pattern string matching algorithms (Aho-Corasick algo. '75) in C (pyahocorasick)
- Dedicated (sparse) linear algebra routines in C (via Cython) exploiting structure of the matrices (Faugère-Lachartre elim. '10)
- Noncommutative signature-based techniques (in the making)