Certifying Exact Complexity Bounds for Matrix Interpretations*

Jose Divasón (U. Rioja)
Sebastiaan Joosten (U. Innsbruck)
Ondřej Kunčar (TU Munich)
René Thiemann (U. Innsbruck)
Akihisa Yamada (U. Innsbruck)

*This research was supported by the Austrian Science Fund (FWF) project Y757

Term rewriting & complexity

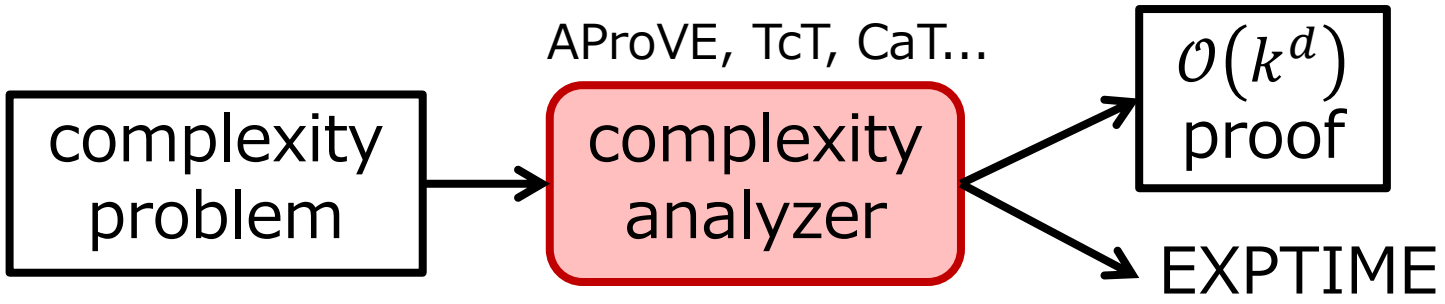
- Example:

```
fun f 0       = s 0
  | f (s 0)   = s 0
  | f (s (s x)) = f (f (s x))
```

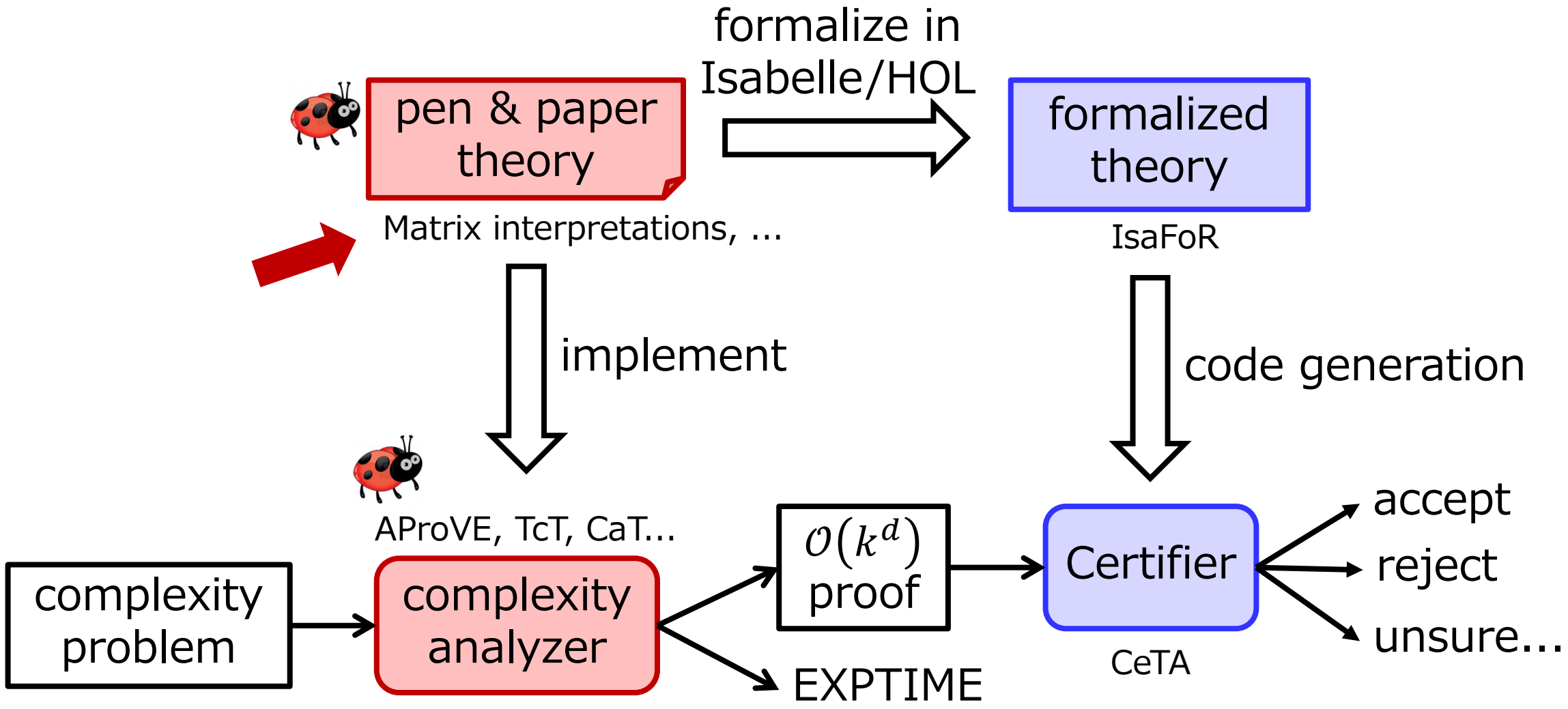
$f(s(s(s\ 0))) \to f(f(s(s\ 0))) \to f(f(f(s\ 0))) \to f(f(s\ 0)) \to f\ (s\ 0) \to s\ 0$

- Complexity problem:

  How long may a rewriting of *"f (s^k 0)"* be?



Certifying complexity proofs



Matrix interpretation [Hofbauer&Waldmann '06, Endrullis+ '08]

- Example:

```
fun f 0         = s 0
  | f (s 0)     = s 0
  | f (s (s x)) = f (f (s x))
```

- Interpret *f, s, 0* as

$$f\ \boldsymbol{x} = \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\boldsymbol{x} + \begin{bmatrix}0\\ 1\end{bmatrix} \qquad s\ \boldsymbol{x} = \begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix}\boldsymbol{x} + \begin{bmatrix}0\\ 1\end{bmatrix} \qquad 0 = \begin{bmatrix}0\\ 0\end{bmatrix}$$

then, $f\big(s(s\ \boldsymbol{x})\big) > f\big(f(s\ \boldsymbol{x})\big)$

runtime of *"f(s^k 0)"* is bounded by $|f(s^k\ x)| \in \mathcal{O}\left(\begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix}^k\right) = \mathcal{O}(k)$

How to check $\mathcal{O}(A^k) = \mathcal{O}(k^d)$ generally?

Checking $A^k \in \mathcal{O}(k^d)$

- let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of A (roots of χ_A)

  if $\exists i.\ |\lambda_i| > 1$, then **reject** (EXPTIME)

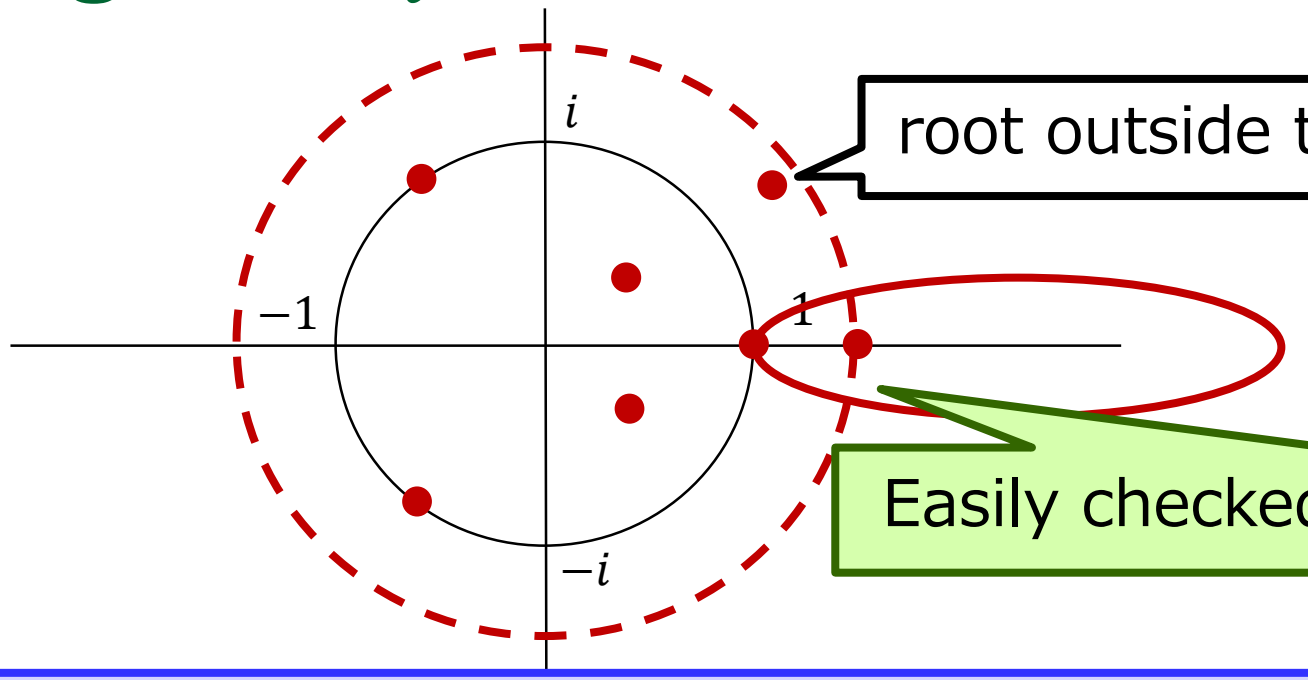
  if $d \geq \dim(A) - 1$, then **accept** [Moser+ '08]

  if $d \geq \max_{|\lambda_i|=1} (\text{multiplicity of } \lambda_i) - 1$, then **accept** [Neureuter+ '11]

  if $d \geq \max_{|\lambda_i|=1} (\text{size of Jordan block for } \lambda_i) - 1$ then **accept** [CPP '16]

  reject (wrong degree)

Checking $\exists i.\ |\lambda_i| > 1$



root outside the circle?

Easily checked via Sturm's method

Theorem (Perron-Forbenius, formalized [AFP]):
If $A \in \mathbb{R}^{n \times n}$ then $\max |\lambda_i|$ is an eigenvalue

Corollary: $\exists i.\ |\lambda_i| > 1$ iff χ_A has a root in $(1, \infty)$

Checking $A^k \in \mathcal{O}(k^d)$

- let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of A (roots of χ_A)

➡ if $\exists i.\ \lambda_i > 1$, then **reject** (EXPTIME)

if $d \geq \dim(A) - 1$, then **accept** [Moser+ '08]

if $d \geq \left(\max_{|\lambda_i|=1} \text{multiplicity of } \lambda_i\right) - 1$, then **accept** [Neureuter+ '11]

if $d \geq \left(\max_{|\lambda_i|=1} \text{size of Jordan block for } \lambda_i\right) - 1$ then **accept** [CPP '16]

reject (wrong degree)

Estimating $\max_{|\lambda_i|=1}(\text{multiplicity of } \lambda_i)$

Proposition: (Yun, formalized [CPP '16])
If Yun's factorization yields $\chi_A = c \cdot f_1^1 \cdot \cdots \cdot f_n^n$,
then f_i and f_j share no root

Corollary:
Maximum multiplicity of any λ_i is n

imprecise, need only $|\lambda_i| = 1$

Checking $A^k \in \mathcal{O}(k^d)$

- let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of A (roots of χ_A)

  if $\exists i.\ \lambda_i > 1$, then **reject** (EXPTIME)

  if $d \geq \dim(A) - 1$, then **accept** [Moser+ '08]

  if $d \geq \max_{|\lambda_i|=1} (\text{multiplicity of } \lambda_i) - 1$, then **accept** [Neureuter+ '11]

  if $d \geq \max_{|\lambda_i|=1} (\text{size of Jordan block for } \lambda_i) - 1$ then **accept** [CPP '16]

  reject (wrong degree)

Jordan normal form (JNF)

- Example:
  - $J = \begin{bmatrix} 2 & 1 & & & \\ & 2 & 1 & & \\ & & 2 & & \\ & & & 3 & 1 \\ & & & & 3 \\ & & & & & 4 \end{bmatrix}$ Jordan blocks, generally $\begin{bmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix}$

Theorem (Jordan, formalized [CPP'16])
Every square matrix has a JNF (over $\mathbb{C}$), i.e. $A = P\,J\,P^{-1}$

Note: $A^k = P\,J\,\boxed{P^{-1}\,P}\,J\,\boxed{P^{-1}}\,\cdots\,\boxed{P}\,J\,P^{-1} = P\,J^k\,P^{-1} \in \mathcal{O}(J^k)$

Power of JNF

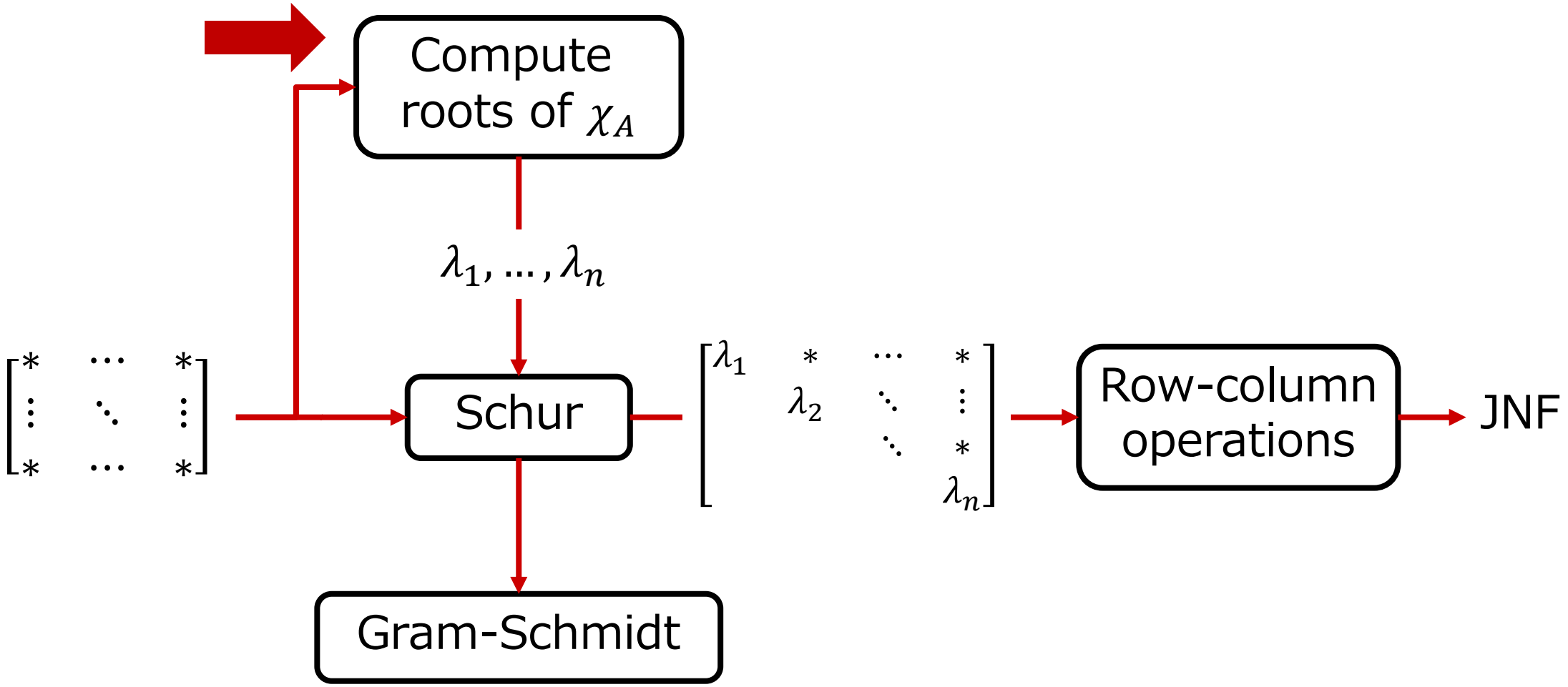
- Example:
  - $J^k = \begin{bmatrix} \boxed{\begin{matrix} 2 & 1 & \\ & 2 & 1 \\ & & 2 \end{matrix}}^{k} & & \\ & \boxed{\begin{matrix} 3 & 1 \\ & 3 \end{matrix}}^{k} & \\ & & \boxed{4}^{k} \end{bmatrix}^{k} = \begin{bmatrix} 2^k & 2^{k-1}k & 2^{k-3}k(k-1) & & \\ & 2^k & 2^{k-1}k & & \\ & & 2^k & & \\ & & & 3^k & 3^{k-1}k \\ & & & & 3^k \\ & & & & & 4^k \end{bmatrix}$

Lemma:

$$\overbrace{\begin{bmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix}}^{\text{size } n}{}^{k} = \begin{bmatrix} \binom{k}{0}\lambda^k & \binom{k}{1}\lambda^{k-1} & \cdots & \binom{k}{n-1}\lambda^{k-n-1} \\ & \binom{k}{0}\lambda^k & \cdots & \binom{k}{n-2}\lambda^{k-n-2} \\ & & \ddots & \vdots \\ & & & \binom{k}{0}\lambda^k \end{bmatrix}$$

polynomially bounded $\Leftrightarrow |\lambda| \leq 1$

Computing JNFs (following [Piziak & Odell '07])



Roots of polynomials

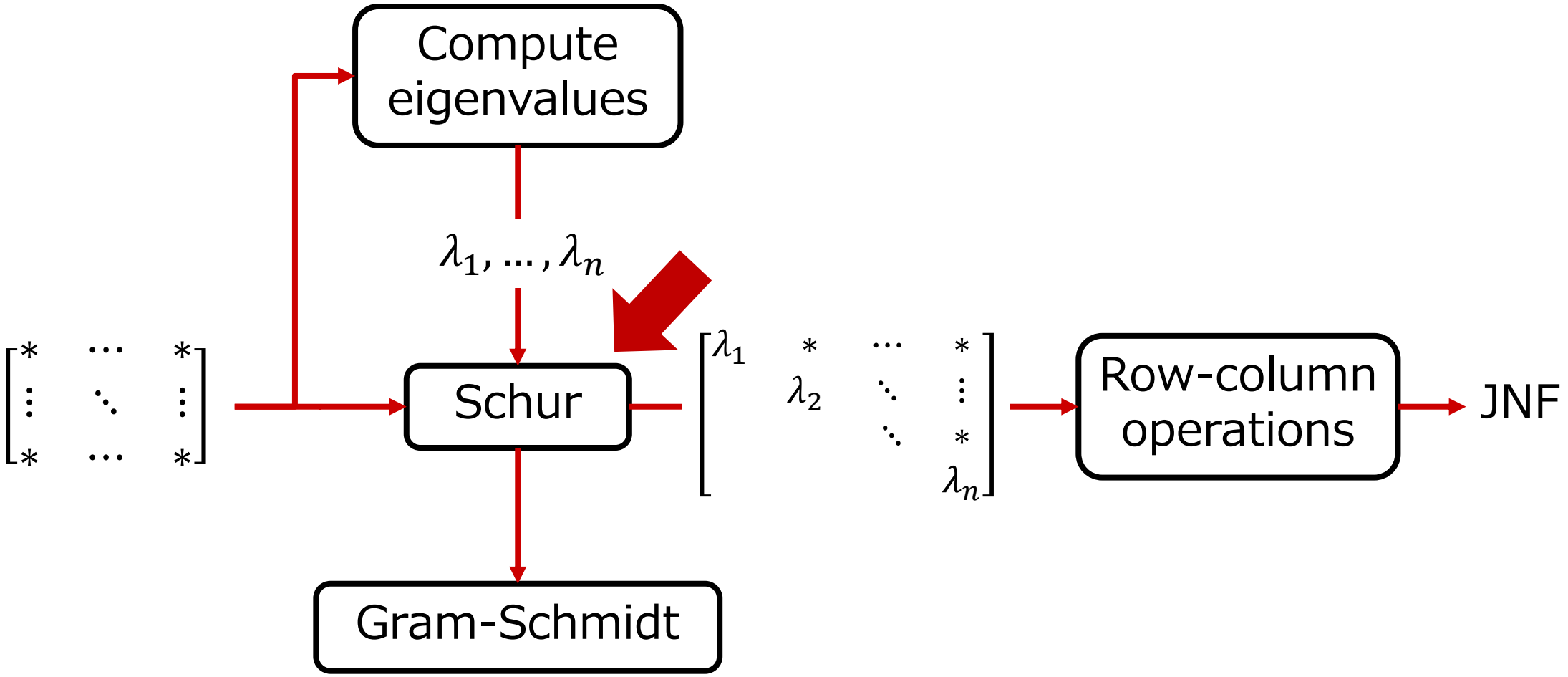
... = algebraic numbers (which we formalized [ITP '16])

- real roots of χ_A are (after factorization)

  "root #1 of χ_A" ... "root #n of χ_A"

- more work for complex roots and norms

Computing JNFs (following [Piziak & Odell '07])



Schur decomposition

Input: $A \in \mathbb{C}^{n \times n}$, eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$

- if $n =$ [illegible] an, ...
- choos[illegible] nogonal basis
- get o[illegible] t)
- compute $V^{-1} A V = \begin{bmatrix} \lambda_1 & b_2 & \cdots & b_n \\ 0 & & & \\ \vdots & & A' & \\ 0 & & & \end{bmatrix}$
- Let $\langle B', P' \rangle = Schur(A', \lambda_2, \cdots, \lambda_n)$
- return $\left\langle \begin{bmatrix} \lambda_1 & b_2 & \cdots & b_n \\ 0 & & & \\ \vdots & & B' & \\ 0 & & & \end{bmatrix}, V \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & P' & \\ 0 & & & \end{bmatrix} \right\rangle$

if V is orthonormal, then

$$V^{-1} = V^H$$

but normalization complicates polynomials

Schur decomposition

Lemma:

if $\chi_A = (x-\lambda_1)\cdots(x-\lambda_n)$ and $Schur\ A\ [\lambda_1,\cdots,\lambda_n] = \langle B, P\rangle$,

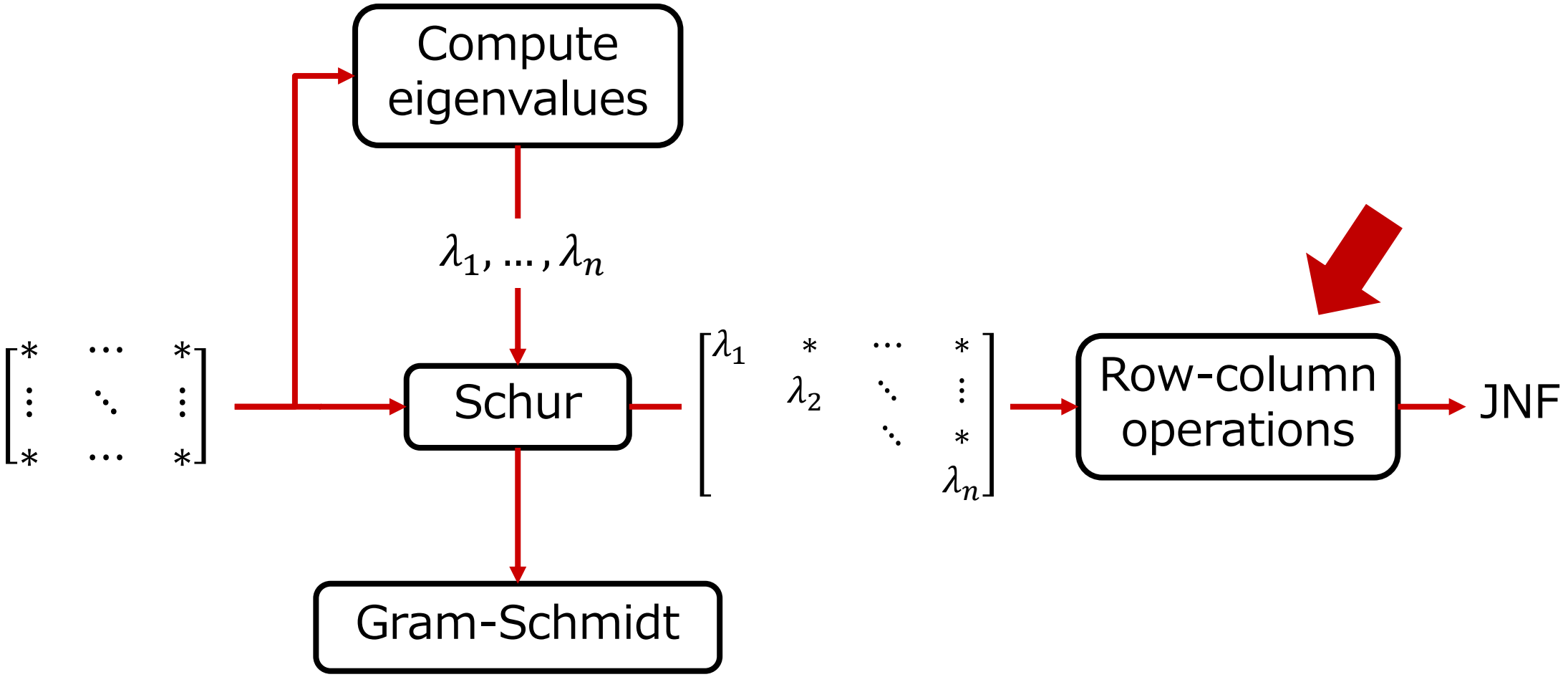
then $A = PBP^{-1}$ and $B = \begin{bmatrix} \lambda_1 & * & \cdots & * \\ & \lambda_2 & \ddots & \vdots \\ & & \ddots & * \\ & & & \lambda_n \end{bmatrix}$

Theorem (Schur, formalized [CPP'16]):

For $\mathbb{C}$ matrix A there exists B s.t.

$A = PBP^{-1}$ and $B = \begin{bmatrix} \lambda_1 & * & \cdots & * \\ & \lambda_2 & \ddots & \vdots \\ & & \ddots & * \\ & & & \lambda_n \end{bmatrix}$

Computing JNFs (following [Piziak & Odell '07])



Column-row operations

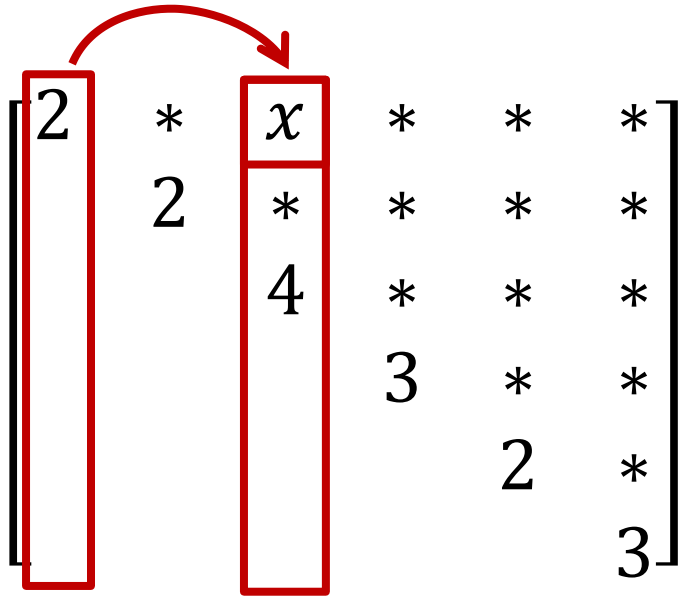
- Elementary similarity preserving operations
  - $A \rightarrow PAP^{-1}$
- Example: *"add_col_sub_row a i j"*

  add $a \times$ column i to column j

  subtract $a \times$ row j from row i

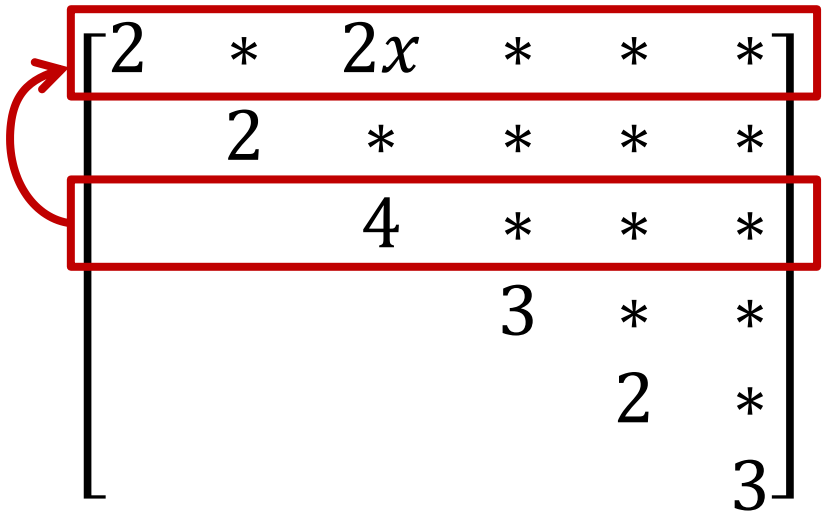
Step 1

add_row_sub_column $\left(\frac{x}{4-2}\right)$



Step 1

add_row_sub_column $\left(\frac{x}{4-2}\right)$



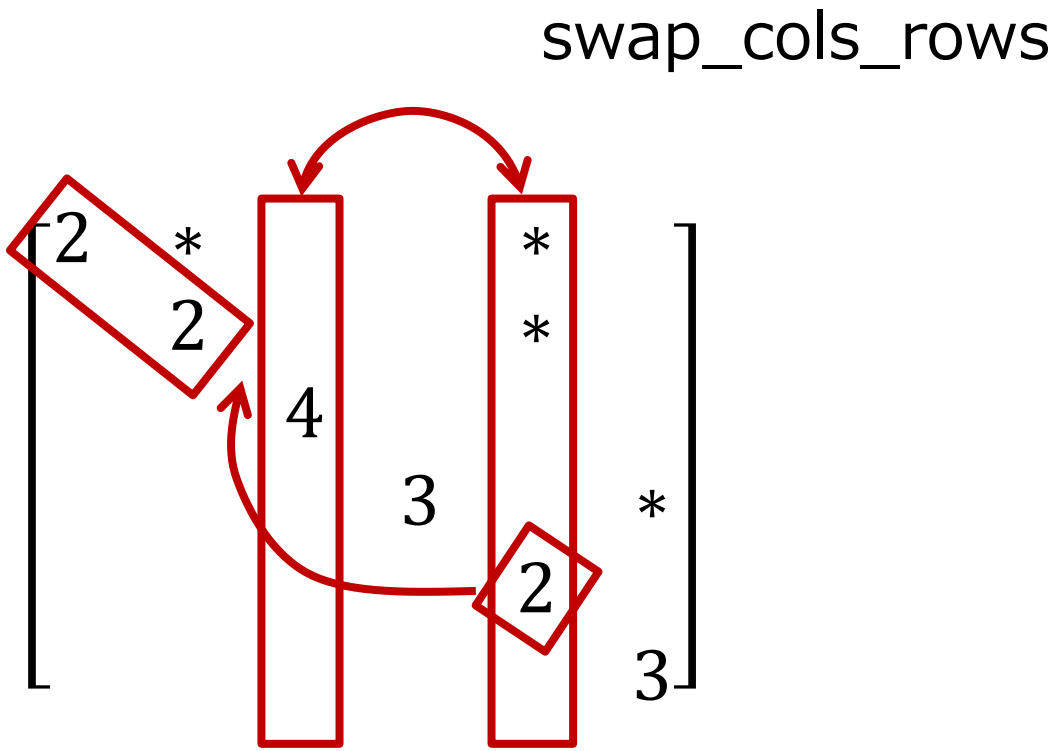
Step 1

$$\begin{bmatrix} 2 & * & \square & * & * & * \\ & 2 & * & * & * & * \\ & & 4 & * & * & * \\ & & & 3 & * & * \\ & & & & 2 & * \\ & & & & & 3 \end{bmatrix}$$

Step 1, end

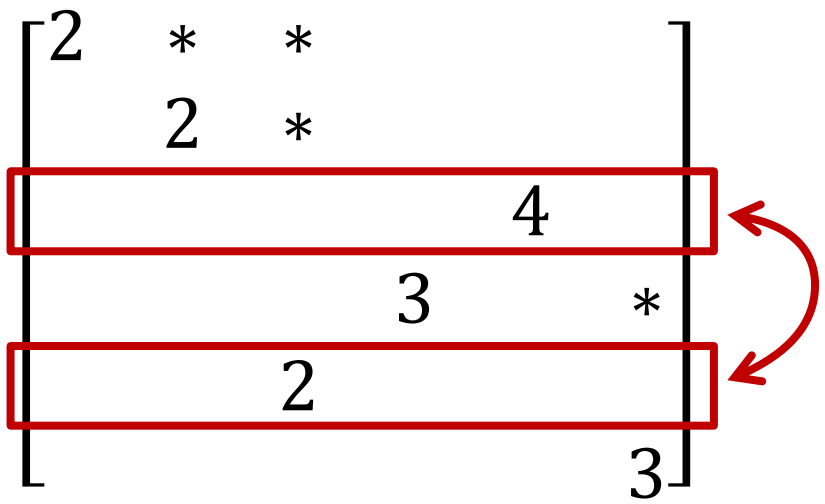
$$\begin{bmatrix} 2 & * & & & * & \\ & 2 & & & * & \\ & & 4 & & & \\ & & & 3 & & * \\ & & & & 2 & \\ & & & & & 3 \end{bmatrix}$$

Step 2 (new)

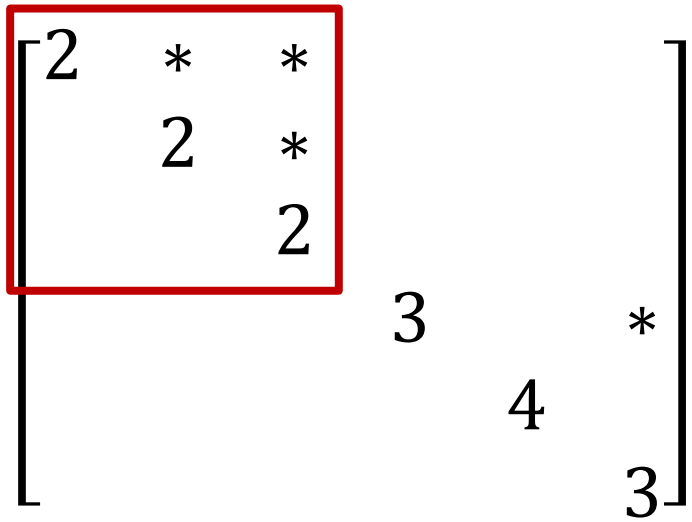


Step 2 (new)

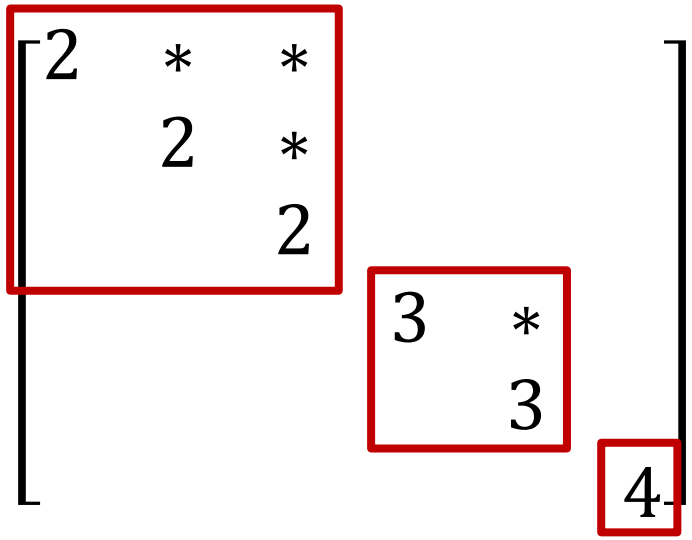
swap_cols_rows



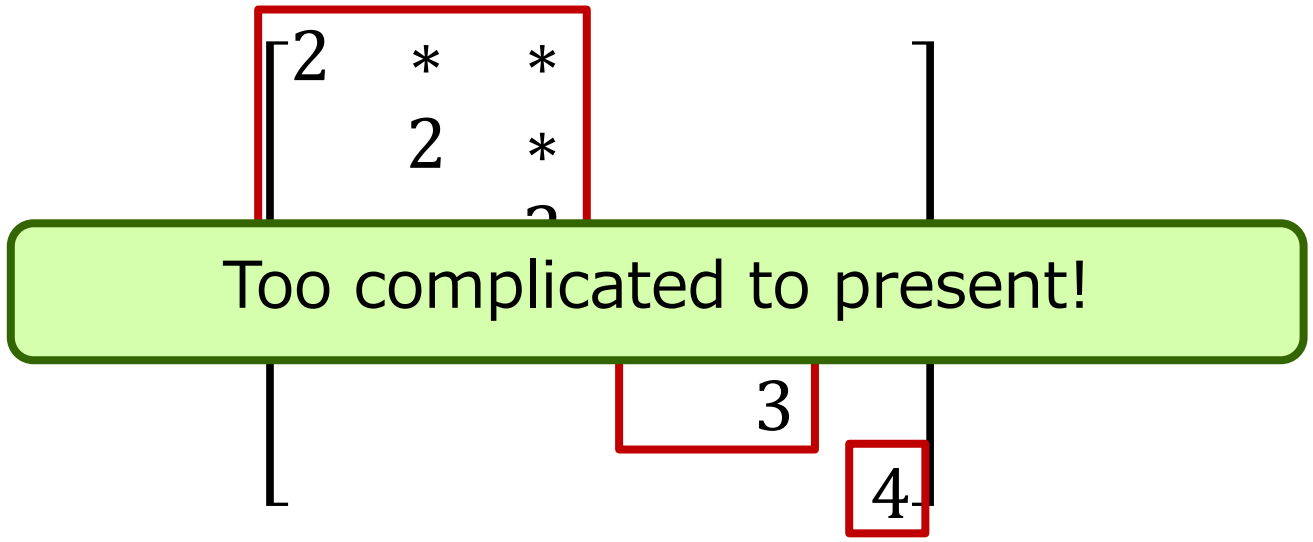
Step 2 (new)



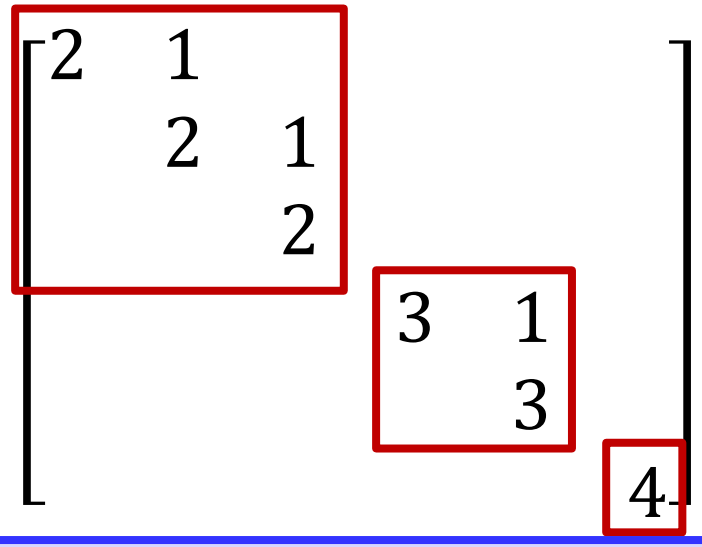
Step 2, finished



Step 3 (main)



Step 3, finished



Lemma: JNF exists for any upper triangular matrix

Theorem (Jordan, formalized [CPP'16]):
JNF exists for any $\mathbb{C}$ matrix

Checking $A^k \in \mathcal{O}(k^d)$

- let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of A (roots of χ_A)

  if $\exists i.\ \lambda_i > 1$, then **reject** (EXPTIME)

  if $d \geq \dim(A) - 1$, then **accept** [Moser+ '08]

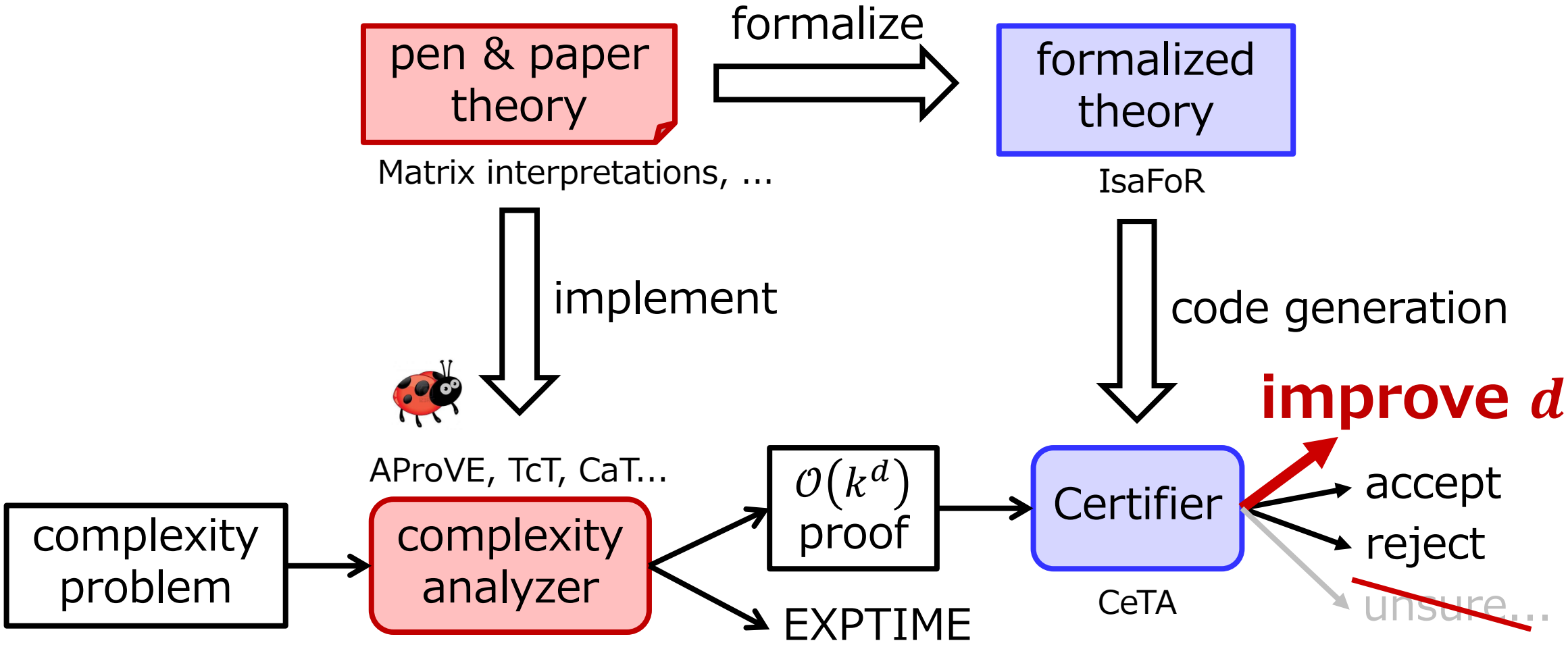
  if $d \geq \max_{|\lambda_i|=1} (\text{multiplicity of } \lambda_i) - 1$, then **accept** [Neureuter+ '11]

  if $d \geq \max_{|\lambda_i|=1} (\text{size of Jordan block for } \lambda_i) - 1$ then **accept** [CPP '16]

  reject (wrong degree)

We are now precise, thanks to JNF

Final picture



Experiments with certifier CeTA

	CaT				TCT				total	
	old	imp.	new	imp.	old	imp.	new	imp.	old	imp.
$\mathcal{O}(1)$	0	0	1	1	0	1	1	1	0	1
$\mathcal{O}(k)$	51	83	99	105	46	68	68	68	51	106
$\mathcal{O}(k^2)$	178	202	212	215	160	173	173	174	178	215
$\mathcal{O}(k^3)$	204	219	223	227	177	185	185	186	206	227
$\mathcal{O}(k^4)$	224	224	232	232	184	186	187	187	224	232
$\mathcal{O}(k^5)$	224	224	232	232	186	186	187	187	224	232

old : upper triangular only [Moser+ '08] (already supported)
new : spectral radius [Neurauter+ '10] or [Middeldorp+ '11]
++ : CeTA improves bounds

Contributions

- polynomial factorization
  - Yun factorization [CPP '16]
  - Berlecamp factorization + Hensel lifting [ongoing]
- Jordan normal form [CPP '16]
  - Matrix library, column-row operations
  - Schur decomposition, Gram-Schmidt orthogonalization
- Algebraic numbers [ITP '16]
- available at

  http://cl-informatik.uibk.ac.at/software/ceta/

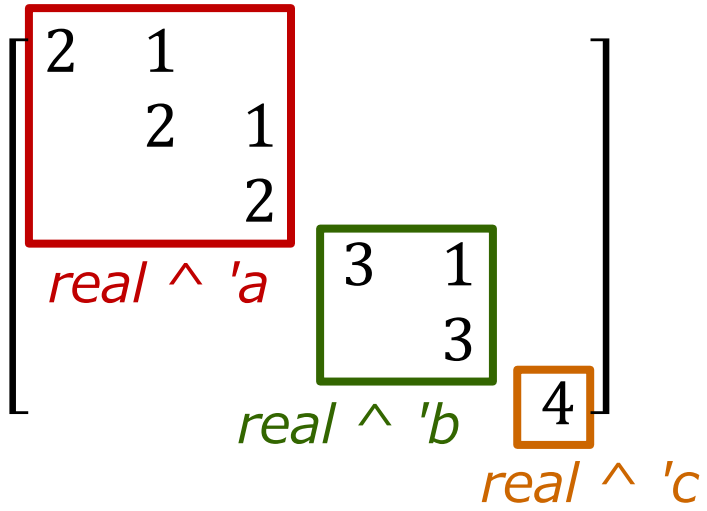
  https://www.isa-afp.org/

Matrix representation in Isabelle

HMA (HOL Multivariate Analysis)

- vector $\approx$ function : $\alpha \Rightarrow \mathbb{R}$
  - α is a finite type with d-elements (d is the dimension)
- dimensions by type inference
  - *"0 + v = v"* meaning *"(0 :: real ^ 'a) + (v :: real ^ 'a) = v"*
  - Nice not for us...

    cannot compose list of block matrices



A new vector/matrix library

- vector $\approx$ dimension $\times$ function : $\mathbb{N} \Rightarrow \mathbb{R}$
  - *"vec d (λ i. v_i)"* = $[v_0, \ldots, v_{d-1}]$
- dimensions explicitly
  - *"dim_v v = d* $\Rightarrow$ $\boldsymbol{0}_v$ *d* $\oplus_v$ *v = v"* meaning $\mathbf{0} + \boldsymbol{v} = \boldsymbol{v}$
  - Suitable for JNFs

"jordan_matrix [(2,3), (3,2), (4,1)]" =

$$\begin{bmatrix} \begin{matrix} 2 & 1 & \\ & 2 & 1 \\ & & 2 \end{matrix} & \begin{matrix} 3 & 1 \\ & 3 \end{matrix} & \begin{matrix} 4 \end{matrix} \end{bmatrix}$$

real mat *real mat* *real mat*

Computing JNFs