

Certification of Termination

Adam Koprowski

Eindhoven University of Technology
Department of Mathematics and Computer Science

24 May 2007

TeReSe

1 CoLoR

- Motivation
- CoLoR architecture
- History
- Overview
- Related work
- Certified competition

2 Formalization of matrix interpretations

- Introduction to matrix interpretations
- Monotone algebras
- Matrices
- Matrix interpretations
- Practicalities

1 CoLoR

- Motivation
- CoLoR architecture
- History
- Overview
- Related work
- Certified competition

2 Formalization of matrix interpretations

- Introduction to matrix interpretations
- Monotone algebras
- Matrices
- Matrix interpretations
- Practicalities

- **Certification of results of termination provers.**
- Common proof format for termination provers:
 - common tools (proof presentation, manipulation, dots),
 - control language for provers (integration of tools)
- Extension of proof assistance kernels.

- Certification of results of termination provers.
- **Common proof format for termination provers:**
 - common tools (proof presentation, manipulation, dots),
 - control language for provers (integration of tools)
- Extension of proof assistance kernels.

- Certification of results of termination provers.
- Common proof format for termination provers:
 - common tools (proof presentation, manipulation, dots),
 - control language for provers (integration of tools)
- Extension of proof assistance kernels.

- Certification of results of termination provers.
- Common proof format for termination provers:
 - common tools (proof presentation, manipulation, dots),
 - **control language for provers (integration of tools)**
- Extension of proof assistance kernels.

- Certification of results of termination provers.
- Common proof format for termination provers:
 - common tools (proof presentation, manipulation, dots),
 - control language for provers (integration of tools)
- Extension of proof assistance kernels.

How to certify termination results?

- **Possibility: certification of tools source code.**
⇒ difficult, tool dependent, extra work with every change, ...
- **CoLoR approach:**
 - **TPG:** common format for termination proofs.
 - Tools output proofs in TPG format.
 - **CoLoR:** a Coq library of results on termination.
 - **Rainbow:** a tool for translation from proofs in TPG format to Coq proofs, using results from CoLoR.

How to certify termination results?

- Possibility: certification of tools source code.
⇒ difficult, tool dependent, extra work with every change, ...
- **CoLoR approach:**
 - TPG: common format for termination proofs.
 - Tools output proofs in TPG format.
 - CoLoR: a Coq library of results on termination.
 - Rainbow: a tool for translation from proofs in TPG format to Coq proofs, using results from CoLoR.

How to certify termination results?

- Possibility: certification of tools source code.
⇒ difficult, tool dependent, extra work with every change, ...
- CoLoR approach:
 - **TPG: common format for termination proofs.**
 - Tools output proofs in TPG format.
 - **CoLoR**: a Coq library of results on termination.
 - **Rainbow**: a tool for translation from proofs in TPG format to Coq proofs, using results from CoLoR.

How to certify termination results?

- Possibility: certification of tools source code.
⇒ difficult, tool dependent, extra work with every change, ...
- CoLoR approach:
 - TPG: common format for termination proofs.
 - Tools output proofs in TPG format.
 - CoLoR: a Coq library of results on termination.
 - Rainbow: a tool for translation from proofs in TPG format to Coq proofs, using results from CoLoR.

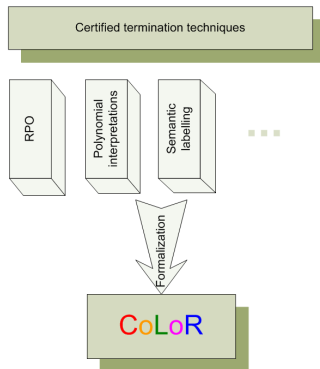
How to certify termination results?

- Possibility: certification of tools source code.
⇒ difficult, tool dependent, extra work with every change, ...
- CoLoR approach:
 - TPG: common format for termination proofs.
 - Tools output proofs in TPG format.
 - CoLoR: a Coq library of results on termination.
 - Rainbow: a tool for translation from proofs in TPG format to Coq proofs, using results from CoLoR.

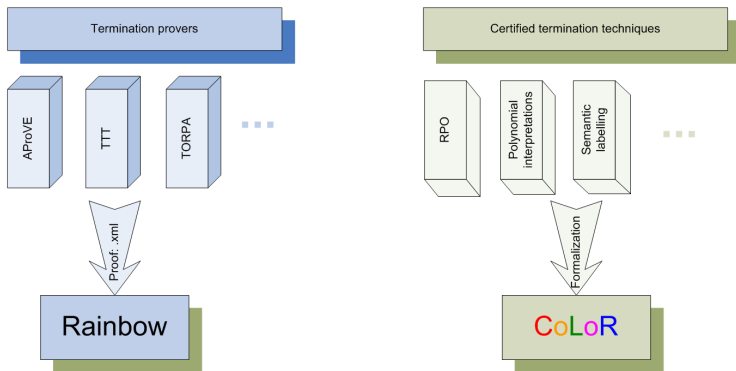
How to certify termination results?

- Possibility: certification of tools source code.
⇒ difficult, tool dependent, extra work with every change, ...
- CoLoR approach:
 - TPG: common format for termination proofs.
 - Tools output proofs in TPG format.
 - CoLoR: a Coq library of results on termination.
 - Rainbow: a tool for translation from proofs in TPG format to Coq proofs, using results from CoLoR.

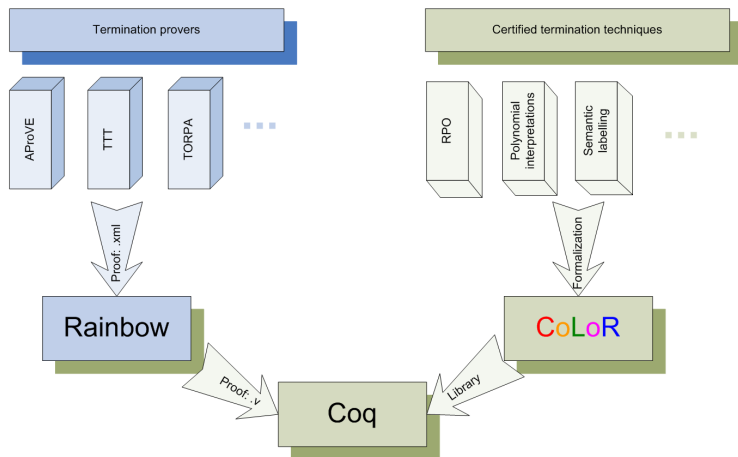
CoLoR architecture overview



CoLoR architecture overview



CoLoR architecture overview



- **Project started (Blanqui)** **March 2004**
- First release **March 2005**
- First certified proofs **July 2006**
- First certification workshop **May 2007**
- First certified competition **June 2007**

- Project started (Blanqui) March 2004
- **First release** March 2005
- First certified proofs July 2006
- First certification workshop May 2007
- First certified competition June 2007

- Project started (Blanqui) March 2004
- First release March 2005
- **First certified proofs** July 2006
- First certification workshop May 2007
- First certified competition June 2007

- Project started (Blanqui) March 2004
- First release March 2005
- First certified proofs July 2006
- **First certification workshop** **May 2007**
- First certified competition June 2007

- Project started (Blanqui) March 2004
- First release March 2005
- First certified proofs July 2006
- First certification workshop May 2007
- **First certified competition June 2007**

- **Termination criteria:**

- **matrix interpretations** [Koprowski, Zantema]
- dependency graph cycles [Blanqui]
- higher-order recursive path ordering [Koprowski]
- recursive path ordering [Coupet-Grimal, Delobel]
- multiset ordering [Koprowski]
- **polynomial interpretations** [Hinderer]

- Transformation techniques:

- dependency pairs [Blanqui]
- rule elimination [Blanqui]
- arguments filtering [Blanqui]
- conversion from algebraic to varyadic terms [Blanqui]

- Termination criteria:
 - **matrix interpretations** [Koprowski, Zantema]
 - dependency graph cycles [Blanqui]
 - higher-order recursive path ordering [Koprowski]
 - recursive path ordering [Coupet-Grimal, Delobel]
 - multiset ordering [Koprowski]
 - **polynomial interpretations** [Hinderer]
- Transformation techniques:
 - dependency pairs [Blanqui]
 - rule elimination [Blanqui]
 - arguments filtering [Blanqui]
 - conversion from algebraic to varyadic terms [Blanqui]

- Termination criteria:
 - **matrix interpretations** [Koprowski, Zantema]
 - **dependency graph cycles** [Blanqui]
 - higher-order recursive path ordering [Koprowski]
 - recursive path ordering [Coupet-Grimal, Delobel]
 - multiset ordering [Koprowski]
 - **polynomial interpretations** [Hinderer]
- Transformation techniques:
 - dependency pairs [Blanqui]
 - rule elimination [Blanqui]
 - arguments filtering [Blanqui]
 - conversion from algebraic to varyadic terms [Blanqui]

- Termination criteria:
 - **matrix interpretations** [Koprowski, Zantema]
 - dependency graph cycles [Blanqui]
 - **higher-order recursive path ordering** [Koprowski]
 - recursive path ordering [Coupet-Grimal, Delobel]
 - multiset ordering [Koprowski]
 - **polynomial interpretations** [Hinderer]
- Transformation techniques:
 - dependency pairs [Blanqui]
 - rule elimination [Blanqui]
 - arguments filtering [Blanqui]
 - conversion from algebraic to varyadic terms [Blanqui]

- Termination criteria:
 - **matrix interpretations** [Koprowski, Zantema]
 - dependency graph cycles [Blanqui]
 - higher-order recursive path ordering [Koprowski]
 - **recursive path ordering** [Coupet-Grimal, Delobel]
 - multiset ordering [Koprowski]
 - **polynomial interpretations** [Hinderer]
- Transformation techniques:
 - dependency pairs [Blanqui]
 - rule elimination [Blanqui]
 - arguments filtering [Blanqui]
 - conversion from algebraic to varyadic terms [Blanqui]

- Termination criteria:
 - **matrix interpretations** [Koprowski, Zantema]
 - dependency graph cycles [Blanqui]
 - higher-order recursive path ordering [Koprowski]
 - recursive path ordering [Coupet-Grimal, Delobel]
 - **multiset ordering** [Koprowski]
 - polynomial interpretations [Hinderer]
- Transformation techniques:
 - dependency pairs [Blanqui]
 - rule elimination [Blanqui]
 - arguments filtering [Blanqui]
 - conversion from algebraic to varyadic terms [Blanqui]

- Termination criteria:
 - **matrix interpretations** [Koprowski, Zantema]
 - dependency graph cycles [Blanqui]
 - higher-order recursive path ordering [Koprowski]
 - recursive path ordering [Coupet-Grimal, Delobel]
 - multiset ordering [Koprowski]
 - **polynomial interpretations** [Hinderer]
- Transformation techniques:
 - dependency pairs [Blanqui]
 - rule elimination [Blanqui]
 - arguments filtering [Blanqui]
 - conversion from algebraic to varyadic terms [Blanqui]

- Termination criteria:
 - **matrix interpretations** [Koprowski, Zantema]
 - dependency graph cycles [Blanqui]
 - higher-order recursive path ordering [Koprowski]
 - recursive path ordering [Coupet-Grimal, Delobel]
 - multiset ordering [Koprowski]
 - **polynomial interpretations** [Hinderer]
- Transformation techniques:
 - dependency pairs [Blanqui]
 - rule elimination [Blanqui]
 - arguments filtering [Blanqui]
 - conversion from algebraic to varyadic terms [Blanqui]

- Termination criteria:
 - **matrix interpretations** [Koprowski, Zantema]
 - dependency graph cycles [Blanqui]
 - higher-order recursive path ordering [Koprowski]
 - recursive path ordering [Coupet-Grimal, Delobel]
 - multiset ordering [Koprowski]
 - **polynomial interpretations** [Hinderer]
- Transformation techniques:
 - **dependency pairs** [Blanqui]
 - rule elimination [Blanqui]
 - arguments filtering [Blanqui]
 - conversion from algebraic to varyadic terms [Blanqui]

- Termination criteria:
 - **matrix interpretations** [Koprowski, Zantema]
 - dependency graph cycles [Blanqui]
 - higher-order recursive path ordering [Koprowski]
 - recursive path ordering [Coupet-Grimal, Delobel]
 - multiset ordering [Koprowski]
 - **polynomial interpretations** [Hinderer]
- Transformation techniques:
 - dependency pairs [Blanqui]
 - **rule elimination** [Blanqui]
 - arguments filtering [Blanqui]
 - conversion from algebraic to varyadic terms [Blanqui]

- Termination criteria:
 - **matrix interpretations** [Koprowski, Zantema]
 - dependency graph cycles [Blanqui]
 - higher-order recursive path ordering [Koprowski]
 - recursive path ordering [Coupet-Grimal, Delobel]
 - multiset ordering [Koprowski]
 - **polynomial interpretations** [Hinderer]
- Transformation techniques:
 - dependency pairs [Blanqui]
 - rule elimination [Blanqui]
 - **arguments filtering** [Blanqui]
 - conversion from algebraic to varyadic terms [Blanqui]

- Termination criteria:
 - **matrix interpretations** [Koprowski, Zantema]
 - dependency graph cycles [Blanqui]
 - higher-order recursive path ordering [Koprowski]
 - recursive path ordering [Coupet-Grimal, Delobel]
 - multiset ordering [Koprowski]
 - **polynomial interpretations** [Hinderer]
- Transformation techniques:
 - dependency pairs [Blanqui]
 - rule elimination [Blanqui]
 - arguments filtering [Blanqui]
 - **conversion from algebraic to varyadic terms** [Blanqui]

- **General libraries:**

- matrices [Koprowski]
- simply typed lambda-terms [Koprowski]
- finite multisets [Koprowski]
- varyadic terms [Blanqui]
- algebraic terms with symbols of fixed arity [Hinderer, Blanqui]
- integer polynomials with multiple variables [Hinderer]
- vectors [Hinderer, Blanqui]
- lists, relations, etc.

- General libraries:

- **matrices [Koprowski]**
- simply typed lambda-terms [Koprowski]
- finite multisets [Koprowski]
- varyadic terms [Blanqui]
- algebraic terms with symbols of fixed arity [Hinderer, Blanqui]
- integer polynomials with multiple variables [Hinderer]
- vectors [Hinderer, Blanqui]
- lists, relations, etc.

- General libraries:
 - matrices [Koprowski]
 - **simply typed lambda-terms [Koprowski]**
 - finite multisets [Koprowski]
 - varyadic terms [Blanqui]
 - algebraic terms with symbols of fixed arity [Hinderer, Blanqui]
 - integer polynomials with multiple variables [Hinderer]
 - vectors [Hinderer, Blanqui]
 - lists, relations, etc.

- General libraries:
 - matrices [Koprowski]
 - simply typed lambda-terms [Koprowski]
 - **finite multisets [Koprowski]**
 - varyadic terms [Blanqui]
 - algebraic terms with symbols of fixed arity [Hinderer, Blanqui]
 - integer polynomials with multiple variables [Hinderer]
 - vectors [Hinderer, Blanqui]
 - lists, relations, etc.

- General libraries:
 - matrices [Koprowski]
 - simply typed lambda-terms [Koprowski]
 - finite multisets [Koprowski]
 - **varyadic terms [Blanqui]**
 - algebraic terms with symbols of fixed arity [Hinderer, Blanqui]
 - integer polynomials with multiple variables [Hinderer]
 - vectors [Hinderer, Blanqui]
 - lists, relations, etc.

- General libraries:
 - matrices [Koprowski]
 - simply typed lambda-terms [Koprowski]
 - finite multisets [Koprowski]
 - varyadic terms [Blanqui]
 - algebraic terms with symbols of fixed arity [Hinderer, Blanqui]
 - integer polynomials with multiple variables [Hinderer]
 - vectors [Hinderer, Blanqui]
 - lists, relations, etc.

- General libraries:
 - matrices [Koprowski]
 - simply typed lambda-terms [Koprowski]
 - finite multisets [Koprowski]
 - varyadic terms [Blanqui]
 - algebraic terms with symbols of fixed arity [Hinderer, Blanqui]
 - integer polynomials with multiple variables [Hinderer]
 - vectors [Hinderer, Blanqui]
 - lists, relations, etc.

- General libraries:
 - matrices [Koprowski]
 - simply typed lambda-terms [Koprowski]
 - finite multisets [Koprowski]
 - varyadic terms [Blanqui]
 - algebraic terms with symbols of fixed arity [Hinderer, Blanqui]
 - integer polynomials with multiple variables [Hinderer]
 - **vectors [Hinderer, Blanqui]**
 - lists, relations, etc.

- General libraries:
 - matrices [Koprowski]
 - simply typed lambda-terms [Koprowski]
 - finite multisets [Koprowski]
 - varyadic terms [Blanqui]
 - algebraic terms with symbols of fixed arity [Hinderer, Blanqui]
 - integer polynomials with multiple variables [Hinderer]
 - vectors [Hinderer, Blanqui]
 - **lists, relations, etc.**

Size of CoLoR

- 42.000 lines of code.
- half of the size of Coq standard library.
- 5% of Coq contribs.

Structure:

- | | |
|---------------------------|-----|
| ● Terms | 44% |
| ● Data structures | 29% |
| ● Termination criteria | 17% |
| ● Mathematical structures | 10% |

Coq constructs:

- | | |
|-----------------------------|------|
| ● Inductive definitions | 38 |
| ● Recursive functions | 116 |
| ● Non-recursive definitions | 560 |
| ● Lemmas and theorems | 2170 |

Size of CoLoR

- 42.000 lines of code.
- half of the size of Coq standard library.
- 5% of Coq contribs.

Structure:

- | | |
|---------------------------|-----|
| ● Terms | 44% |
| ● Data structures | 29% |
| ● Termination criteria | 17% |
| ● Mathematical structures | 10% |

Coq constructs:

- | | |
|-----------------------------|------|
| ● Inductive definitions | 38 |
| ● Recursive functions | 116 |
| ● Non-recursive definitions | 560 |
| ● Lemmas and theorems | 2170 |

Size of CoLoR

- 42.000 lines of code.
- half of the size of Coq standard library.
- 5% of Coq contribs.

Structure:

- | | |
|---------------------------|-----|
| ● Terms | 44% |
| ● Data structures | 29% |
| ● Termination criteria | 17% |
| ● Mathematical structures | 10% |

Coq constructs:

- | | |
|-----------------------------|------|
| ● Inductive definitions | 38 |
| ● Recursive functions | 116 |
| ● Non-recursive definitions | 560 |
| ● Lemmas and theorems | 2170 |

Size of CoLoR

- 42.000 lines of code.
- half of the size of Coq standard library.
- 5% of Coq contribs.

Structure:

- **Terms** 44%
- Data structures 29%
- Termination criteria 17%
- Mathematical structures 10%

Coq constructs:

- Inductive definitions 38
- Recursive functions 116
- Non-recursive definitions 560
- Lemmas and theorems 2170

Size of CoLoR

- 42.000 lines of code.
- half of the size of Coq standard library.
- 5% of Coq contribs.

Structure:

- Terms 44%
- **Data structures** 29%
- Termination criteria 17%
- Mathematical structures 10%

Coq constructs:

- Inductive definitions 38
- Recursive functions 116
- Non-recursive definitions 560
- Lemmas and theorems 2170

Size of CoLoR

- 42.000 lines of code.
- half of the size of Coq standard library.
- 5% of Coq contribs.

Structure:

- Terms 44%
- Data structures 29%
- Termination criteria 17%
- Mathematical structures 10%

Coq constructs:

- Inductive definitions 38
- Recursive functions 116
- Non-recursive definitions 560
- Lemmas and theorems 2170

Size of CoLoR

- 42.000 lines of code.
- half of the size of Coq standard library.
- 5% of Coq contribs.

Structure:

- | | |
|----------------------------------|------------|
| • Terms | 44% |
| • Data structures | 29% |
| • Termination criteria | 17% |
| • Mathematical structures | 10% |

Coq constructs:

- | | |
|-----------------------------|------|
| • Inductive definitions | 38 |
| • Recursive functions | 116 |
| • Non-recursive definitions | 560 |
| • Lemmas and theorems | 2170 |

Size of CoLoR

- 42.000 lines of code.
- half of the size of Coq standard library.
- 5% of Coq contribs.

Structure:

- | | |
|---------------------------|-----|
| • Terms | 44% |
| • Data structures | 29% |
| • Termination criteria | 17% |
| • Mathematical structures | 10% |

Coq constructs:

- | | |
|--------------------------------|-----------|
| • Inductive definitions | 38 |
| • Recursive functions | 116 |
| • Non-recursive definitions | 560 |
| • Lemmas and theorems | 2170 |

Size of CoLoR

- 42.000 lines of code.
- half of the size of Coq standard library.
- 5% of Coq contribs.

Structure:

- | | |
|---------------------------|-----|
| • Terms | 44% |
| • Data structures | 29% |
| • Termination criteria | 17% |
| • Mathematical structures | 10% |

Coq constructs:

- | | |
|------------------------------|------------|
| • Inductive definitions | 38 |
| • Recursive functions | 116 |
| • Non-recursive definitions | 560 |
| • Lemmas and theorems | 2170 |

Size of CoLoR

- 42.000 lines of code.
- half of the size of Coq standard library.
- 5% of Coq contribs.

Structure:

- | | |
|---------------------------|-----|
| • Terms | 44% |
| • Data structures | 29% |
| • Termination criteria | 17% |
| • Mathematical structures | 10% |

Coq constructs:

- | | |
|------------------------------------|------------|
| • Inductive definitions | 38 |
| • Recursive functions | 116 |
| • Non-recursive definitions | 560 |
| • Lemmas and theorems | 2170 |

Size of CoLoR

- 42.000 lines of code.
- half of the size of Coq standard library.
- 5% of Coq contribs.

Structure:

- | | |
|---------------------------|-----|
| • Terms | 44% |
| • Data structures | 29% |
| • Termination criteria | 17% |
| • Mathematical structures | 10% |

Coq constructs:

- | | |
|-----------------------------|------|
| • Inductive definitions | 38 |
| • Recursive functions | 116 |
| • Non-recursive definitions | 560 |
| • Lemmas and theorems | 2170 |

- **CoLoR project**
Authors: Blanqui, ...
Tool: TPA, ...
Proof assistant: Coq
- **A3PAT project**
Authors: Contejean, ...
Tool: CiME
Proof assistant: Coq
- **Isabelle/HOL termination checker**
Authors: Bulwahn, Krauss, Nipkow, ...
Tool: TTT
Proof assistant: Isabelle/HOL

- CoLoR project

Authors: Blanqui, ...

Tool: TPA, ...

Proof assistant: Coq

- A3PAT project

Authors: Contejean, ...

Tool: CiME

Proof assistant: Coq

- Isabelle/HOL termination checker

Authors: Bulwahn, Krauss, Nipkow, ...

Tool: T_TT

Proof assistant: Isabelle/HOL

- **CoLoR project**

Authors: Blanqui, ...

Tool: TPA, ...

Proof assistant: Coq

- **A3PAT project**

Authors: Contejean, ...

Tool: CiME

Proof assistant: Coq

- **Isabelle/HOL termination checker**

Authors: Bulwahn, Krauss, Nipkow, ...

Tool: T_TT

Proof assistant: Isabelle/HOL

- In the termination competition this year a new “certified” category introduced.
- Participants:
 - CiME + A3PAT
 - TPA + CoLoR
 - T_1T_2 + CoLoR
 - AProVE + A3PAT (?)
- Many questions remain, like
 - Who's the winner?
 - Competition VS Cooperation

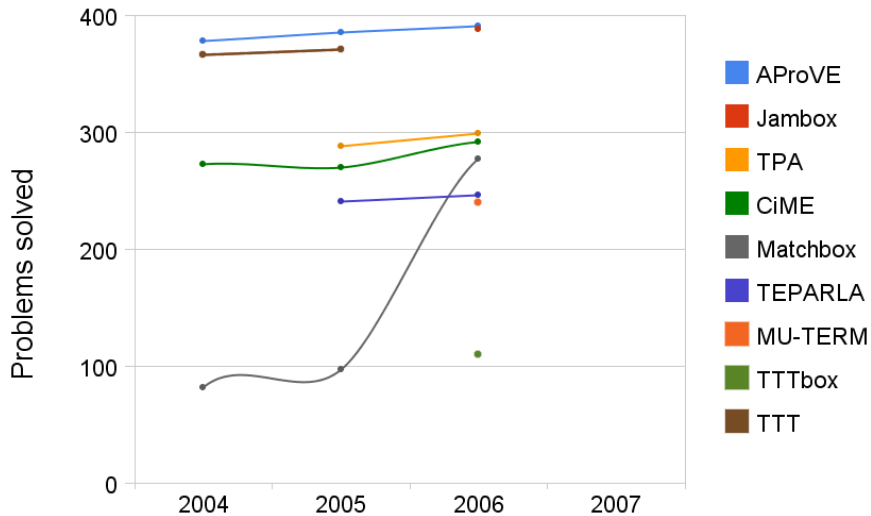
- In the termination competition this year a new “certified” category introduced.
- **Participants:**
 - CiME + A3PAT
 - TPA + CoLoR
 - $T\bar{T}_2$ + CoLoR
 - AProVE + A3PAT (?)
- Many questions remain, like
 - Who's the winner?
 - Competition VS Cooperation

- In the termination competition this year a new “certified” category introduced.
- Participants:
 - CiME + A3PAT
 - TPA + CoLoR
 - $T_T T_2$ + CoLoR
 - AProVE + A3PAT (?)
- Many questions remain, like
 - Who's the winner?
 - Competition VS Cooperation

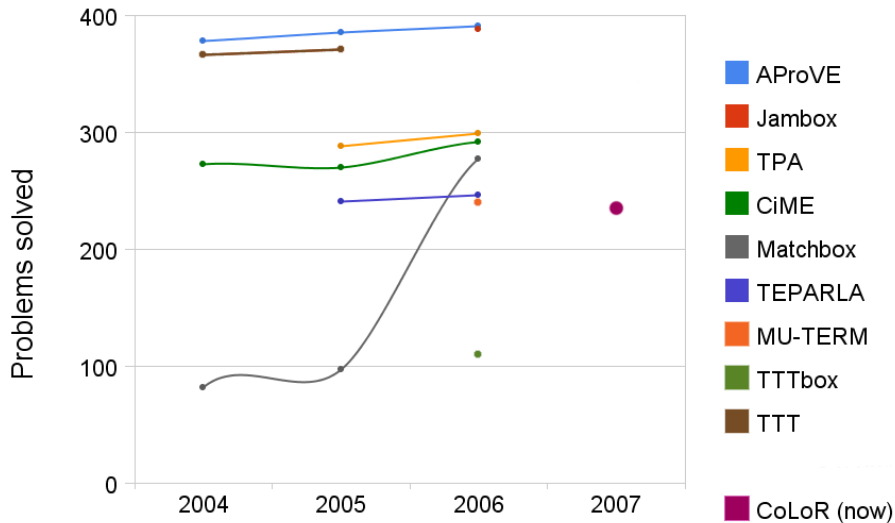
- In the termination competition this year a new “certified” category introduced.
- Participants:
 - CiME + A3PAT
 - TPA + CoLoR
 - $T_T T_2$ + CoLoR
 - AProVE + A3PAT (?)
- Many questions remain, like
 - Who's the winner?
 - Competition VS Cooperation

- In the termination competition this year a new “certified” category introduced.
- Participants:
 - CiME + A3PAT
 - TPA + CoLoR
 - $T_T T_2$ + CoLoR
 - AProVE + A3PAT (?)
- Many questions remain, like
 - Who's the winner?
 - Competition VS Cooperation

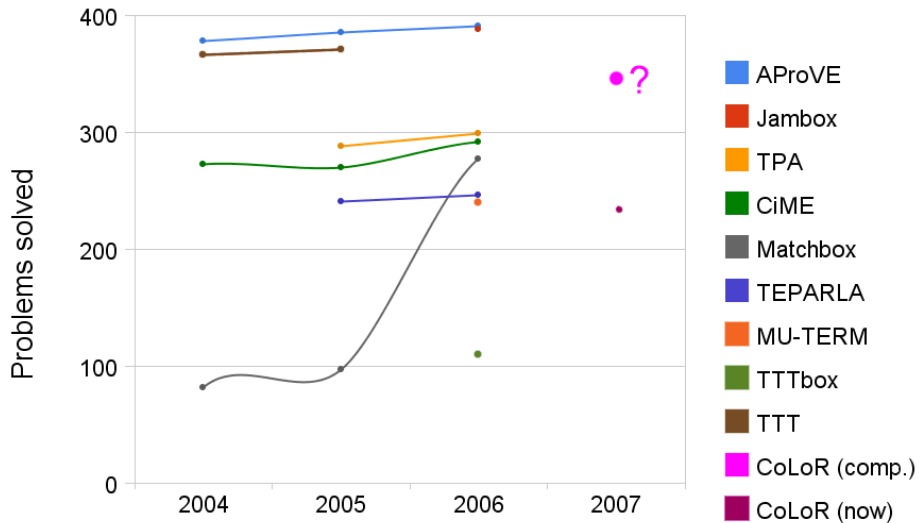
Termination competition



Termination competition



Termination competition



1 CoLoR

- Motivation
- CoLoR architecture
- History
- Overview
- Related work
- Certified competition

2 Formalization of matrix interpretations

- Introduction to matrix interpretations
- Monotone algebras
- Matrices
- Matrix interpretations
- Practicalities

Example

z086.trs

$a(a(x)) \rightarrow c(b(x)), \quad b(b(x)) \rightarrow c(a(x)), \quad c(c(x)) \rightarrow b(a(x))$

Matrix interpretation for z086.trs

$$a(x) = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix}$$

$$b(x) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$c(x) = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Example

z086.trs

$a(a(x)) \rightarrow c(b(x)), \quad b(b(x)) \rightarrow c(a(x)), \quad c(c(x)) \rightarrow b(a(x))$

Matrix interpretation for z086.trs

$$a(x) = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix}$$

$$b(x) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$c(x) = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Termination proof for z086.trs

$$a(a(x)) = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix}$$

$$c(b(x)) = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Example ctd.

Termination proof for z086.trs

$$\begin{aligned} a(a(x)) &= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix} \\ c(b(x)) &= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix} \end{aligned}$$

Definition (Monotonicity)

An operation $[f] : A \times \dots \times A \rightarrow A$ is *monotone* with respect to a binary relation \triangleright on A if

$$a_i \triangleright a'_i \implies [f](a_1, \dots, a_i, \dots, a_n) \triangleright [f](a_1, \dots, a'_i, \dots, a_n).$$

Definition

Given a relation \triangleright on A we define its extension to a relation on terms as:

$$s \triangleright_{\mathcal{T}} t \equiv \forall \alpha : \mathcal{X} \rightarrow A, [s, \alpha] \triangleright [t, \alpha]$$

Definition (Monotonicity)

An operation $[f] : A \times \dots \times A \rightarrow A$ is *monotone* with respect to a binary relation \triangleright on A if

$$a_i \triangleright a'_i \implies [f](a_1, \dots, a_i, \dots, a_n) \triangleright [f](a_1, \dots, a'_i, \dots, a_n).$$

Definition

Given a relation \triangleright on A we define its extension to a relation on terms as:

$$s \triangleright_{\mathcal{T}} t \equiv \forall \alpha : \mathcal{X} \rightarrow A, [s, \alpha] \triangleright [t, \alpha]$$

Definition (A weakly monotone Σ -algebra)

A *weakly monotone Σ -algebra* $(A, [\cdot], >, \succsim)$ is a Σ -algebra $(A, [\cdot])$ equipped with two binary relations $>, \succsim$ on A such that

- $>$ is well-founded;
- $> \cdot \succsim \subseteq >$;
- for every $f \in \Sigma$ the operation $[f]$ is monotone with respect to \succsim .

Definition (An *extended monotone Σ -algebra*)

An *extended monotone Σ -algebra* $(A, [\cdot], >, \succsim)$ is a weakly monotone Σ -algebra $(A, [\cdot], >, \succsim)$ in which moreover for every $f \in \Sigma$ the operation $[f]$ is monotone with respect to $>$.

Definition (A weakly monotone Σ -algebra)

A *weakly monotone Σ -algebra* $(A, [\cdot], >, \gtrsim)$ is a Σ -algebra $(A, [\cdot])$ equipped with two binary relations $>, \gtrsim$ on A such that

- $>$ is well-founded;
- $> \cdot \gtrsim \subseteq >$;
- for every $f \in \Sigma$ the operation $[f]$ is monotone with respect to \gtrsim .

Definition (An *extended monotone Σ -algebra*)

An *extended monotone Σ -algebra* $(A, [\cdot], >, \gtrsim)$ is a weakly monotone Σ -algebra $(A, [\cdot], >, \gtrsim)$ in which moreover for every $f \in \Sigma$ the operation $[f]$ is monotone with respect to $>$.

Theorem

Let R, R', S, S' be TRSs over a signature Σ , $(A, [\cdot], >, \succeq)$ be an extended monotone Σ -algebra such that:

- $\ell \succeq_{\mathcal{T}} r$ for every rule $\ell \rightarrow r$ in $R \cup S$ and
- $\ell >_{\mathcal{T}} r$ for every rule $\ell \rightarrow r$ in $R' \cup S'$

Then $\text{SN}(R/S)$ implies $\text{SN}(R \cup R' / S \cup S')$.

Theorem

Let R, R', S, S' be TRSs over a signature Σ , let $(A, [\cdot], >, \succeq)$ be a weakly monotone Σ -algebra such that:

- $\ell \succeq_{\mathcal{T}} r$ for every rule $\ell \rightarrow r$ in $R \cup S$ and
- $\ell >_{\mathcal{T}} r$ for every rule $\ell \rightarrow r$ in R' ,

Then $\text{SN}(R_{\text{top}}/S)$ implies $\text{SN}((R \cup R')_{\text{top}}/S)$.

Theorem

Let R, R', S, S' be TRSs over a signature Σ , $(A, [\cdot], >, \gtrsim)$ be an extended monotone Σ -algebra such that:

- $\ell \gtrsim_{\mathcal{T}} r$ for every rule $\ell \rightarrow r$ in $R \cup S$ and
- $\ell >_{\mathcal{T}} r$ for every rule $\ell \rightarrow r$ in $R' \cup S'$

Then $\text{SN}(R/S)$ implies $\text{SN}(R \cup R' / S \cup S')$.

Theorem

Let R, R', S, S' be TRSs over a signature Σ , let $(A, [\cdot], >, \gtrsim)$ be a weakly monotone Σ -algebra such that:

- $\ell \gtrsim_{\mathcal{T}} r$ for every rule $\ell \rightarrow r$ in $R \cup S$ and
- $\ell >_{\mathcal{T}} r$ for every rule $\ell \rightarrow r$ in R' ,

Then $\text{SN}(R_{\text{top}}/S)$ implies $\text{SN}((R \cup R')_{\text{top}}/S)$.

Formalization of monotone algebras

- **Monotone algebras are formalized as a functor.**
- Apart for the aforementioned requirements there is one additional required to deal with concrete examples: $>_{\mathcal{T}}$ and $\gtrsim_{\mathcal{T}}$ must be decidable.
- More precisely the requirement is to provide a relation \gg , such that
 - $\gg \subseteq >_{\mathcal{T}}$ and
 - \gg is decidable
 - similarly for \gtrsim .
- The structure returned by the functor contains all the machinery required to prove (relative)-(top)-termination in Coq.

Formalization of monotone algebras

- Monotone algebras are formalized as a functor.
- Apart for the aforementioned requirements there is one additional required to deal with concrete examples: $>_{\mathcal{T}}$ and $\gtrsim_{\mathcal{T}}$ must be decidable.
- More precisely the requirement is to provide a relation \gg , such that
 - $\gg \subseteq >_{\mathcal{T}}$ and
 - \gg is decidable
 - similarly for \gtrsim .
- The structure returned by the functor contains all the machinery required to prove (relative)-(top)-termination in Coq.

Formalization of monotone algebras

- Monotone algebras are formalized as a functor.
- Apart for the aforementioned requirements there is one additional required to deal with concrete examples: $>_{\mathcal{T}}$ and $\gtrsim_{\mathcal{T}}$ must be decidable.
- More precisely the requirement is to provide a relation \gg , such that
 - $\gg \subseteq >_{\mathcal{T}}$ and
 - \gg is decidable
 - similarly for \gtrsim .
- The structure returned by the functor contains all the machinery required to prove (relative)-(top)-termination in Coq.

Formalization of monotone algebras

- Monotone algebras are formalized as a functor.
- Apart for the aforementioned requirements there is one additional required to deal with concrete examples: $>_{\mathcal{T}}$ and $\gtrsim_{\mathcal{T}}$ must be decidable.
- More precisely the requirement is to provide a relation \gg , such that
 - $\gg \subseteq >_{\mathcal{T}}$ and
 - \gg is decidable
 - similarly for \gtrsim .
- The structure returned by the functor contains all the machinery required to prove (relative)-(top)-termination in Coq.

Formalization of matrices

- Matrices are formalized as a functor taking as an argument the semi-ring of coefficients \mathcal{R} and providing a structure of matrices of arbitrary sizes with coefficients in \mathcal{R} and
- a number of basic operations over matrices such as:

$$[\cdot], \quad M_{i,j}, \quad M + N, \quad M * N, \quad M^T, \dots$$

- and a number of basic properties such as:
 - $M + N = N + M$,
 - $M * (N * P) = (M * N) * P$
 - monotonicity of $*$
 - ...

Formalization of matrices

- Matrices are formalized as a functor taking as an argument the semi-ring of coefficients \mathcal{R} and providing a structure of matrices of arbitrary sizes with coefficients in \mathcal{R} and
- a number of basic operations over matrices such as:

$$[\cdot], \quad M_{i,j}, \quad M + N, \quad M * N, \quad M^T, \dots$$

- and a number of basic properties such as:
 - $M + N = N + M$,
 - $M * (N * P) = (M * N) * P$
 - monotonicity of $*$
 - ...

Formalization of matrices

- Matrices are formalized as a functor taking as an argument the semi-ring of coefficients \mathcal{R} and providing a structure of matrices of arbitrary sizes with coefficients in \mathcal{R} and
- a number of basic operations over matrices such as:

$$[\cdot], \quad M_{i,j}, \quad M + N, \quad M * N, \quad M^T, \dots$$

- and a number of basic properties such as:
 - $M + N = N + M$,
 - $M * (N * P) = (M * N) * P$
 - monotonicity of $*$
 - ...

Polynomial interpretations in the setting of monotone algebras

- $A = \mathbb{Z}$,
- $> = >_{\mathbb{Z}}$, $\gtrsim = \geq_{\mathbb{Z}}$,
- interpretations represented by polynomials
 $[f(x_1, \dots, x_n)] = P_{\mathbb{Z}}(x_1, \dots, x_n)$,
- $>_{\mathcal{T}}$ not decidable (positiveness of polynomial) — heuristics required.

Polynomial interpretations in the setting of monotone algebras

- $A = \mathbb{Z}$,
- $> = >_{\mathbb{Z}}$, $\gtrsim = \geq_{\mathbb{Z}}$,
- interpretations represented by polynomials
 $[f(x_1, \dots, x_n)] = P_{\mathbb{Z}}(x_1, \dots, x_n)$,
- $>_{\mathcal{T}}$ not decidable (positiveness of polynomial) — heuristics required.

Polynomial interpretations in the setting of monotone algebras

- $A = \mathbb{Z}$,
- $> = >_{\mathbb{Z}}$, $\gtrsim = \geq_{\mathbb{Z}}$,
- **interpretations represented by polynomials**
 $[f(x_1, \dots, x_n)] = P_{\mathbb{Z}}(x_1, \dots, x_n)$,
- $>_{\mathcal{T}}$ not decidable (positiveness of polynomial) — heuristics required.

Polynomial interpretations in the setting of monotone algebras

- $A = \mathbb{Z}$,
- $> = >_{\mathbb{Z}}$, $\gtrsim = \geq_{\mathbb{Z}}$,
- interpretations represented by polynomials
 $[f(x_1, \dots, x_n)] = P_{\mathbb{Z}}(x_1, \dots, x_n)$,
- $>_{\mathcal{T}}$ not decidable (positiveness of polynomial) — heuristics required.

Matrix interpretations in the setting of monotone algebras

- **fix a dimension d ,**
- $A = \mathbb{N}^d$,
- $(u_1, \dots, u_d) \succeq (v_1, \dots, v_d)$ iff $\forall i, u_i \geq_{\mathbb{N}} v_i$,
- $(u_1, \dots, u_d) > (v_1, \dots, v_d)$ iff $(u_1, \dots, u_d) \succeq (v_1, \dots, v_d) \wedge u_1 >_{\mathbb{N}} v_1$,
- interpretations represented as:
 $[f(x_1, \dots, x_n)] = M_1 x_1 + \dots + M_n x_n + v$
where $M_i \in \mathbb{N}^{d \times d}$, $v \in \mathbb{N}^d$,
- $>_{\mathcal{T}}$ and $\succeq_{\mathcal{T}}$ are decidable in this case but thanks to introducing \gg we do not need to prove completeness of their characterization.
- Domain fixed to \mathbb{N} with natural orders $>$ and \geq .

Matrix interpretations in the setting of monotone algebras

- fix a dimension d ,
- $A = \mathbb{N}^d$,
- $(u_1, \dots, u_d) \succeq (v_1, \dots, v_d)$ iff $\forall i, u_i \geq_{\mathbb{N}} v_i$,
- $(u_1, \dots, u_d) > (v_1, \dots, v_d)$ iff $(u_1, \dots, u_d) \succeq (v_1, \dots, v_d) \wedge u_1 >_{\mathbb{N}} v_1$,
- interpretations represented as:
 $[f(x_1, \dots, x_n)] = M_1 x_1 + \dots + M_n x_n + v$
where $M_i \in \mathbb{N}^{d \times d}$, $v \in \mathbb{N}^d$,
- $>_{\mathcal{T}}$ and $\succeq_{\mathcal{T}}$ are decidable in this case but thanks to introducing \gg we do not need to prove completeness of their characterization.
- Domain fixed to \mathbb{N} with natural orders $>$ and \geq .

Matrix interpretations in the setting of monotone algebras

- fix a dimension d ,
- $A = \mathbb{N}^d$,
- $(u_1, \dots, u_d) \succsim (v_1, \dots, v_d)$ iff $\forall i, u_i \geq_{\mathbb{N}} v_i$,
- $(u_1, \dots, u_d) > (v_1, \dots, v_d)$ iff $(u_1, \dots, u_d) \succsim (v_1, \dots, v_d) \wedge u_1 >_{\mathbb{N}} v_1$,
- interpretations represented as:
 $[f(x_1, \dots, x_n)] = M_1 x_1 + \dots + M_n x_n + v$
where $M_i \in \mathbb{N}^{d \times d}$, $v \in \mathbb{N}^d$,
- $>_{\mathcal{T}}$ and $\succsim_{\mathcal{T}}$ are decidable in this case but thanks to introducing \gg we do not need to prove completeness of their characterization.
- Domain fixed to \mathbb{N} with natural orders $>$ and \geq .

Matrix interpretations in the setting of monotone algebras

- fix a dimension d ,
- $A = \mathbb{N}^d$,
- $(u_1, \dots, u_d) \succeq (v_1, \dots, v_d)$ iff $\forall i, u_i \geq_{\mathbb{N}} v_i$,
- $(u_1, \dots, u_d) > (v_1, \dots, v_d)$ iff $(u_1, \dots, u_d) \succeq (v_1, \dots, v_d) \wedge u_1 >_{\mathbb{N}} v_1$,
- interpretations represented as:
 $[f(x_1, \dots, x_n)] = M_1 x_1 + \dots + M_n x_n + v$
where $M_i \in \mathbb{N}^{d \times d}$, $v \in \mathbb{N}^d$,
- $>_{\mathcal{T}}$ and $\succeq_{\mathcal{T}}$ are decidable in this case but thanks to introducing \gg we do not need to prove completeness of their characterization.
- Domain fixed to \mathbb{N} with natural orders $>$ and \geq .

Matrix interpretations in the setting of monotone algebras

- fix a dimension d ,
- $A = \mathbb{N}^d$,
- $(u_1, \dots, u_d) \succeq (v_1, \dots, v_d)$ iff $\forall i, u_i \geq_{\mathbb{N}} v_i$,
- $(u_1, \dots, u_d) > (v_1, \dots, v_d)$ iff $(u_1, \dots, u_d) \succeq (v_1, \dots, v_d) \wedge u_1 >_{\mathbb{N}} v_1$,
- **interpretations represented as:**
 $[f(x_1, \dots, x_n)] = M_1 x_1 + \dots + M_n x_n + v$
where $M_i \in \mathbb{N}^{d \times d}$, $v \in \mathbb{N}^d$,
- $>_{\mathcal{T}}$ and $\succeq_{\mathcal{T}}$ are decidable in this case but thanks to introducing \gg we do not need to prove completeness of their characterization.
- Domain fixed to \mathbb{N} with natural orders $>$ and \geq .

Matrix interpretations in the setting of monotone algebras

- fix a dimension d ,
- $A = \mathbb{N}^d$,
- $(u_1, \dots, u_d) \succeq (v_1, \dots, v_d)$ iff $\forall i, u_i \geq_{\mathbb{N}} v_i$,
- $(u_1, \dots, u_d) > (v_1, \dots, v_d)$ iff $(u_1, \dots, u_d) \succeq (v_1, \dots, v_d) \wedge u_1 >_{\mathbb{N}} v_1$,
- interpretations represented as:
 $[f(x_1, \dots, x_n)] = M_1 x_1 + \dots + M_n x_n + v$
where $M_i \in \mathbb{N}^{d \times d}$, $v \in \mathbb{N}^d$,
- $>_{\mathcal{T}}$ and $\succeq_{\mathcal{T}}$ are decidable in this case but thanks to introducing \gg we do not need to prove completeness of their characterization.
- Domain fixed to \mathbb{N} with natural orders $>$ and \geq .

Matrix interpretations in the setting of monotone algebras

- fix a dimension d ,
- $A = \mathbb{N}^d$,
- $(u_1, \dots, u_d) \succeq (v_1, \dots, v_d)$ iff $\forall i, u_i \geq_{\mathbb{N}} v_i$,
- $(u_1, \dots, u_d) > (v_1, \dots, v_d)$ iff $(u_1, \dots, u_d) \succeq (v_1, \dots, v_d) \wedge u_1 >_{\mathbb{N}} v_1$,
- interpretations represented as:
 $[f(x_1, \dots, x_n)] = M_1 x_1 + \dots + M_n x_n + v$
where $M_i \in \mathbb{N}^{d \times d}$, $v \in \mathbb{N}^d$,
- $>_{\mathcal{T}}$ and $\succeq_{\mathcal{T}}$ are decidable in this case but thanks to introducing \gg we do not need to prove completeness of their characterization.
- Domain fixed to \mathbb{N} with natural orders $>$ and \geq .

Formalization size (LOC):

- Monotone algebras: 351
- Matrices: 642
- Matrix interpretations: 673
- Polynomial interpretations in MA setting: 116

Evaluation of **TPA + Rainbow** on **TPDB 3.2** (864 TRSs):

- **polynomial interpretations:** 167
- matrix interpretations: 237
- polynomial and matrix interpretations: 275
 - Verification time: AVG: 5sec. MAX: 75sec.
 - Certificate size: AVG: 25kB. MAX: 437kB
 - Proof steps: AVG: 5 MAX: 29

Evaluation of **TPA + Rainbow** on **TPDB 3.2** (864 TRSs):

- polynomial interpretations: 167
- **matrix interpretations:** **237**
- polynomial and matrix interpretations: 275
 - Verification time: AVG: 5sec. MAX: 75sec.
 - Certificate size: AVG: 25kB. MAX: 437kB
 - Proof steps: AVG: 5 MAX: 29

Evaluation of **TPA + Rainbow** on **TPDB 3.2** (864 TRSs):

- polynomial interpretations: 167
- matrix interpretations: 237
- **polynomial and matrix interpretations: 275**
 - Verification time: AVG: 5sec. MAX: 75sec.
 - Certificate size: AVG: 25kB. MAX: 437kB
 - Proof steps: AVG: 5 MAX: 29

Evaluation of **TPA + Rainbow** on **TPDB 3.2** (864 TRSs):

- polynomial interpretations: 167
- matrix interpretations: 237
- polynomial and matrix interpretations: 275
 - **Verification time:** **AVG: 5sec.** **MAX: 75sec.**
 - Certificate size: **AVG: 25kB.** **MAX: 437kB**
 - Proof steps: **AVG: 5** **MAX: 29**

Evaluation of **TPA + Rainbow** on **TPDB 3.2** (864 TRSs):

- polynomial interpretations: 167
- matrix interpretations: 237
- polynomial and matrix interpretations: 275
 - Verification time: AVG: 5sec. MAX: 75sec.
 - **Certificate size:** **AVG: 25kB.** **MAX: 437kB**
 - Proof steps: AVG: 5 MAX: 29

Evaluation of **TPA + Rainbow** on **TPDB 3.2** (864 TRSs):

- polynomial interpretations: 167
- matrix interpretations: 237
- polynomial and matrix interpretations: 275
 - Verification time: AVG: 5sec. MAX: 75sec.
 - Certificate size: AVG: 25kB. MAX: 437kB
 - **Proof steps: AVG: 5 MAX: 29**

Figure: Before



Figure: Now



```
http://color.loria.fr
```



Thank you for your attention.