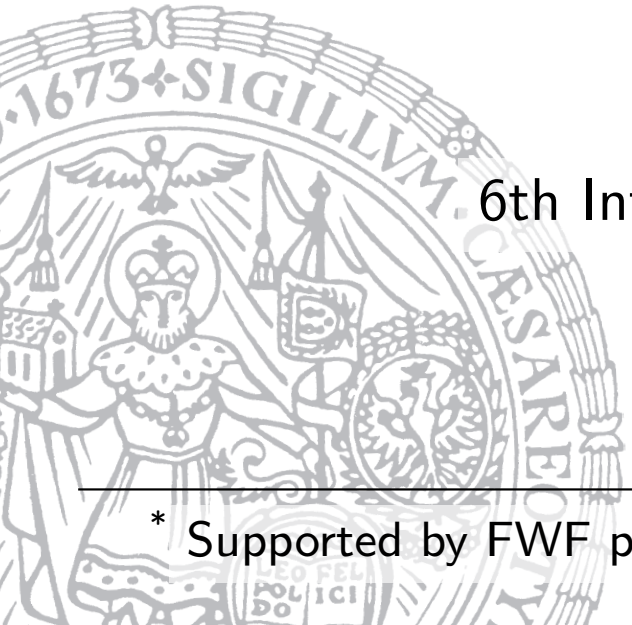
Computational Logic

Aspects of Layer Systems in IsaFoR*

Bertram Felgenhauer and Franziska Rapp

University of Innsbruck



6th International Workshop on Confluence
Oxford, 2017-09-08

* Supported by FWF project P27528

Motivation

- confluence tools prove confluence, but may be buggy
- solution: certification
  - IsaFoR, CeTA

Motivation

- confluence tools prove confluence, but may be buggy
- solution: certification
  - IsaFoR, CeTA
- current state: $\approx 50\%$ of the proofs by CSI are certified
- missing techniques:
  - decomposition based on (order-sorted) persistence
  - development closedness
  - AC techniques
  - advanced labeling techniques
  - ...

Motivation

- confluence tools prove confluence, but may be buggy
- solution: certification
  - IsaFoR, CeTA
- current state: $\approx 50\%$ of the proofs by CSI are certified
- missing techniques:
  - decomposition based on (order-sorted) persistence
  - development closedness
  - AC techniques
  - advanced labeling techniques
  - ...

$\Rightarrow$ goal: formalize persistence & co.

Table of Contents

- Introduction
- Layer Systems
- Formalization

Modularity

Theorem (Toyama 1987, Klop *et al.* 1994, van Oostrom 2008, . . .)

Let $\mathcal{R}_1$, $\mathcal{R}_2$ be TRSs over disjoint signatures. Then

$$\mathsf{CR}(\mathcal{R}_1 \cup \mathcal{R}_2) \iff \mathsf{CR}(\mathcal{R}_1) \wedge \mathsf{CR}(\mathcal{R}_2)$$

Proof idea

- $\Longrightarrow$ is easy (homogeneous terms are closed under rewriting)

Modularity

Theorem (Toyama 1987, Klop *et al.* 1994, van Oostrom 2008, . . .)

Let $\mathcal{R}_1$, $\mathcal{R}_2$ be TRSs over disjoint signatures. Then

$$\mathsf{CR}(\mathcal{R}_1 \cup \mathcal{R}_2) \iff \mathsf{CR}(\mathcal{R}_1) \wedge \mathsf{CR}(\mathcal{R}_2)$$

Proof idea

- $\Longrightarrow$ is easy (homogeneous terms are closed under rewriting)
- recursively decompose terms into homogeneous maximal top and aliens

Modularity

Theorem (Toyama 1987, Klop *et al.* 1994, van Oostrom 2008, . . .)

Let $\mathcal{R}_1$, $\mathcal{R}_2$ be TRSs over disjoint signatures. Then

$$\mathsf{CR}(\mathcal{R}_1 \cup \mathcal{R}_2) \iff \mathsf{CR}(\mathcal{R}_1) \wedge \mathsf{CR}(\mathcal{R}_2)$$

Proof idea

- $\Longrightarrow$ is easy (homogeneous terms are closed under rewriting)
- recursively decompose terms into homogeneous maximal top and aliens
- use induction on rank

Modularity

Theorem (Toyama 1987, Klop *et al.* 1994, van Oostrom 2008, . . .)

Let $\mathcal{R}_1$, $\mathcal{R}_2$ be TRSs over disjoint signatures. Then

$$\mathsf{CR}(\mathcal{R}_1 \cup \mathcal{R}_2) \iff \mathsf{CR}(\mathcal{R}_1) \wedge \mathsf{CR}(\mathcal{R}_2)$$

Proof idea

- $\Longrightarrow$ is easy (homogeneous terms are closed under rewriting)
- recursively decompose terms into homogeneous maximal top and aliens
- use induction on rank
- . . . details are complicated

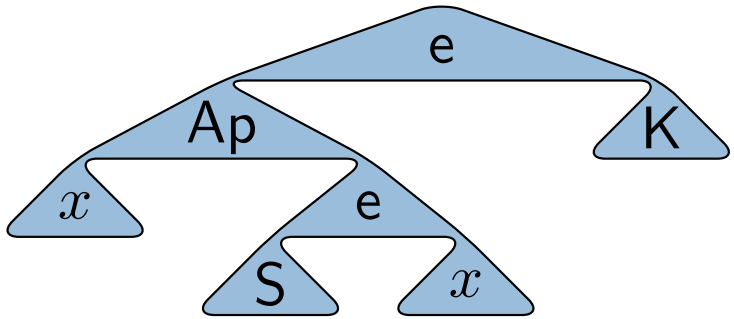
Example

$$\mathcal{R}_1 = \{\, \mathsf{Ap}(\mathsf{Ap}(\mathsf{K}, x), y) \to x,$$
$$\mathsf{Ap}(\mathsf{Ap}(\mathsf{Ap}(\mathsf{S}, x), y), z) \to \mathsf{Ap}(\mathsf{Ap}(x, z), \mathsf{Ap}(y, z)) \,\}$$
$$\mathcal{R}_2 = \{\, \mathsf{e}(x, x) \to \top \,\}$$

Example

$$\mathcal{R}_1 = \{\, \mathsf{Ap}(\mathsf{Ap}(\mathsf{K}, x), y) \to x,$$
$$\mathsf{Ap}(\mathsf{Ap}(\mathsf{Ap}(\mathsf{S}, x), y), z) \to \mathsf{Ap}(\mathsf{Ap}(x, z), \mathsf{Ap}(y, z)) \,\}$$
$$\mathcal{R}_2 = \{\, \mathsf{e}(x, x) \to \top \,\}$$

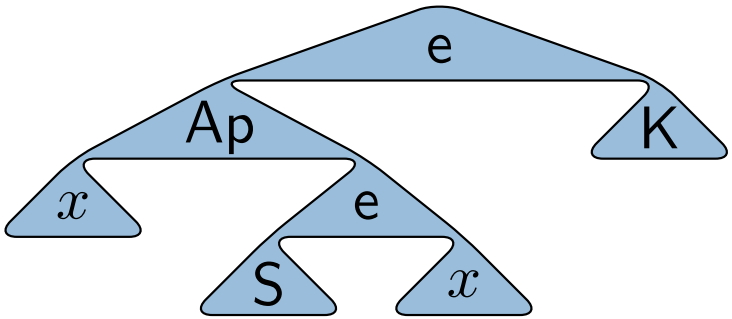
- $\mathsf{e}(\mathsf{Ap}(x, \mathsf{e}(\mathsf{S}, x)), \mathsf{K})$

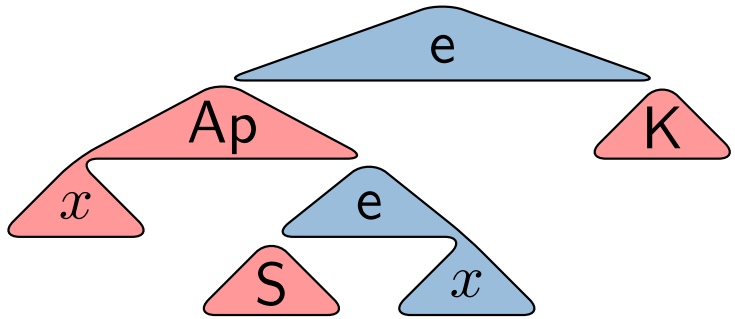


Example

$$\mathcal{R}_1 = \{\, \mathsf{Ap}(\mathsf{Ap}(\mathsf{K}, x), y) \to x,$$
$$\mathsf{Ap}(\mathsf{Ap}(\mathsf{Ap}(\mathsf{S}, x), y), z) \to \mathsf{Ap}(\mathsf{Ap}(x, z), \mathsf{Ap}(y, z)) \,\}$$
$$\mathcal{R}_2 = \{\, \mathsf{e}(x, x) \to \top \,\}$$

- $\mathsf{e}(\mathsf{Ap}(x, \mathsf{e}(\mathsf{S}, x)), \mathsf{K})$

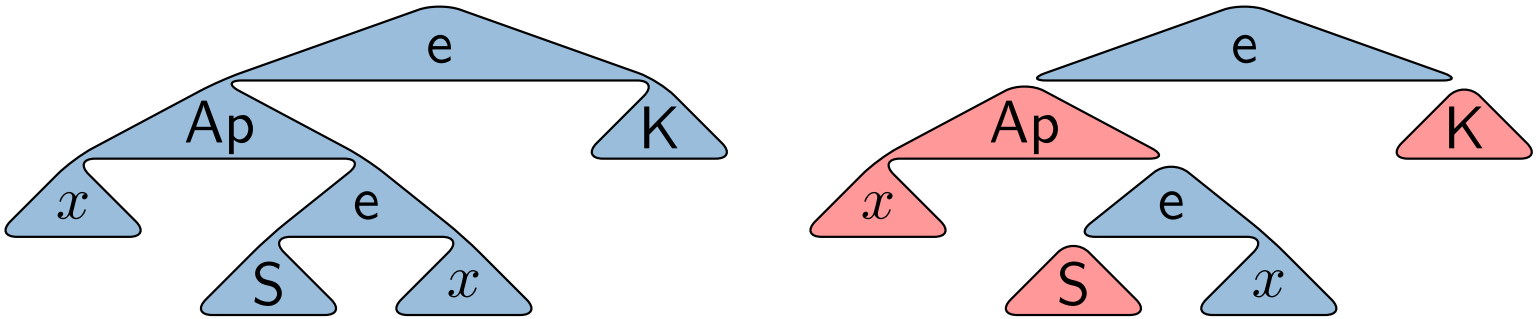




Example

$$\mathcal{R}_1 = \{\, \mathsf{Ap}(\mathsf{Ap}(\mathsf{K}, x), y) \to x,$$
$$\mathsf{Ap}(\mathsf{Ap}(\mathsf{Ap}(\mathsf{S}, x), y), z) \to \mathsf{Ap}(\mathsf{Ap}(x, z), \mathsf{Ap}(y, z)) \,\}$$
$$\mathcal{R}_2 = \{\, \mathsf{e}(x, x) \to \top \,\}$$

- $\mathsf{e}(\mathsf{Ap}(x, \mathsf{e}(\mathsf{S}, x)), \mathsf{K})$

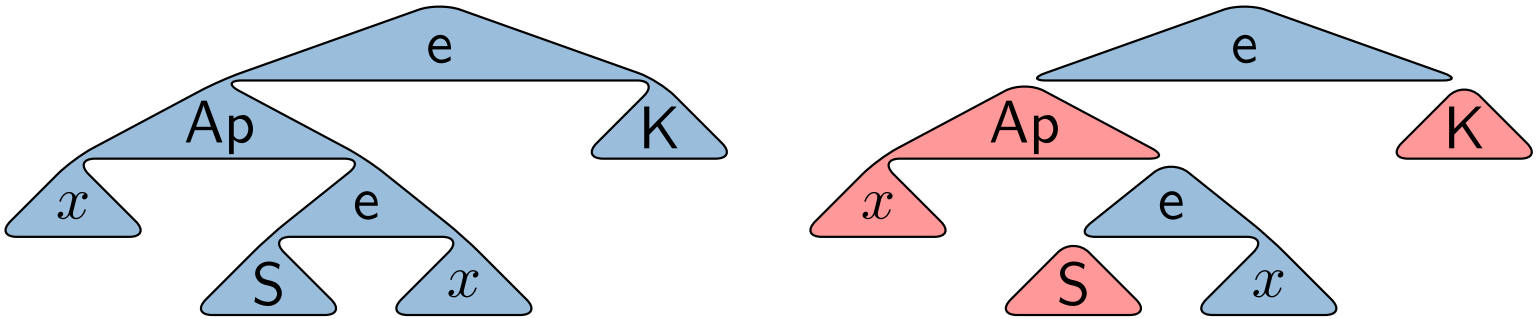


- max-top $\mathsf{e}(\square, \square)$, aliens $\mathsf{Ap}(x, \mathsf{e}(\mathsf{S}, x))$ and K, rank 4

Example

$$\mathcal{R}_1 = \{\, \mathsf{Ap}(\mathsf{Ap}(\mathsf{K}, x), y) \to x,$$
$$\mathsf{Ap}(\mathsf{Ap}(\mathsf{Ap}(\mathsf{S}, x), y), z) \to \mathsf{Ap}(\mathsf{Ap}(x, z), \mathsf{Ap}(y, z)) \,\}$$
$$\mathcal{R}_2 = \{\, \mathsf{e}(x, x) \to \top \,\}$$

- $\mathsf{e}(\mathsf{Ap}(x, \mathsf{e}(\mathsf{S}, x)), \mathsf{K})$



- max-top $\mathsf{e}(\Box, \Box)$, aliens $\mathsf{Ap}(x, \mathsf{e}(\mathsf{S}, x))$ and K, rank 4
- $\mathcal{R}_1$ and $\mathcal{R}_2$ are confluent $\implies$ $\mathcal{R}_1 \cup \mathcal{R}_2$ is confluent

Related Results

Results

- Persistence (Aoto and Toyama 1997)
- Layer preservation (Ohlebusch 1994)
- Quasi-ground systems (Kitahara *et al.* 1995)
- Currying (Kahrs 1995)
- Order-sorted persistence (Aoto and Toyama 1996, F. *et al.* 2015)

Related Results

Results

- Persistence (Aoto and Toyama 1997)
- Layer preservation (Ohlebusch 1994)
- Quasi-ground systems (Kitahara *et al.* 1995)
- Currying (Kahrs 1995)
- Order-sorted persistence (Aoto and Toyama 1996, F. *et al.* 2015)

Proof idea

- $\Longrightarrow$ is easy
- recursively decompose terms into max-top and aliens
- use induction on rank
- . . . details are complicated

Table of Contents

- Introduction
- Layer Systems
- Formalization

Layer Systems for Proving Confluence

Idea

- layer system $\mathfrak{L}$: set of multi-hole contexts
- top of t: $L \in \mathfrak{L}$ with $L \sqsubseteq t$
- max-top, aliens, rank
- notions: layer system, weakly layered, layered

Layer Systems for Proving Confluence

Idea

- layer system $\mathfrak{L}$: set of multi-hole contexts
- top of t: $L \in \mathfrak{L}$ with $L \sqsubseteq t$
- max-top, aliens, rank
- notions: layer system, weakly layered, layered

Main Results

- If $\mathcal{R}$ is layered wrt. $\mathfrak{L}$ and terms of rank 1 are confluent, then $\mathcal{R}$ is confluent.
- ...

Layer Systems for Proving Confluence

Idea

- layer system $\mathfrak{L}$: set of multi-hole contexts
- top of t: $L \in \mathfrak{L}$ with $L \sqsubseteq t$
- max-top, aliens, rank
- notions: layer system, weakly layered, layered

Main Results

- If $\mathcal{R}$ is layered wrt. $\mathfrak{L}$ and terms of rank 1 are confluent, then $\mathcal{R}$ is confluent.
- ...

Applications

- modularity: $\mathcal{R}_1 \cup \mathcal{R}_2$ is layered by $\mathcal{T}(\mathcal{F}_1, \mathcal{V}) \cup \mathcal{T}(\mathcal{F}_2, \mathcal{V})$.
- ...

Layer Systems Definition

Definition

Under the following conditions, the set of contexts $\mathfrak{L} \subseteq \mathcal{C}$ is layer system:

- L_1 Every term in $\mathcal{T}(\mathcal{F}, \mathcal{V})$ has a non-empty top
- L_2 If $x \in \mathcal{V}$ and $C \in \mathcal{C}$, then $C[x]_p \in \mathfrak{L}$ if and only if $C[\Box]_p \in \mathfrak{L}$
- L_3 If $L, N \in \mathfrak{L}$, $p \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(L)$, and $L|_p \sqcup N$ is defined, then $L[L|_p \sqcup N]_p \in \mathfrak{L}$

Layer Systems Definition

Definition

Under the following conditions, the TRS $\mathcal{R}$ is weakly layered wrt. $\mathfrak{L}$:

- L_1 Every term in $\mathcal{T}(\mathcal{F},\mathcal{V})$ has a non-empty top
- L_2 If $x \in \mathcal{V}$ and $C \in \mathcal{C}$, then $C[x]_p \in \mathfrak{L}$ if and only if $C[\Box]_p \in \mathfrak{L}$
- L_3 If $L, N \in \mathfrak{L}$, $p \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(L)$, and $L|_p \sqcup N$ is defined, then $L[L|_p \sqcup N]_p \in \mathfrak{L}$
- W If M is the max-top of s, $p \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(M)$, and $s \to_{p,\ell\to r} t$ with $\ell \to r \in \mathcal{R}$, then $M \to_{p,\ell\to r} L$ for some $L \in \mathfrak{L}$

Layer Systems Definition

Definition

Under the following conditions, the TRS $\mathcal{R}$ is layered wrt. $\mathfrak{L}$:

- L_1 Every term in $\mathcal{T}(\mathcal{F},\mathcal{V})$ has a non-empty top
- L_2 If $x \in \mathcal{V}$ and $C \in \mathcal{C}$, then $C[x]_p \in \mathfrak{L}$ if and only if $C[\Box]_p \in \mathfrak{L}$
- L_3 If $L, N \in \mathfrak{L}$, $p \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(L)$, and $L|_p \sqcup N$ is defined, then $L[L|_p \sqcup N]_p \in \mathfrak{L}$
- W If M is the max-top of s, $p \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(M)$, and $s \to_{p,\ell\to r} t$ with $\ell \to r \in \mathcal{R}$, then $M \to_{p,\ell\to r} L$ for some $L \in \mathfrak{L}$
- C_1 In (W), either L is the max-top of t or $L = \Box$
- C_2 If $L, N \in \mathfrak{L}$ and $L \sqsubseteq N$, then $L[N|_p]_p \in \mathfrak{L}$ for any $p \in \mathcal{P}\mathrm{os}_{\Box}(L)$

Layer Systems Definition

Definition

Under the following conditions, the TRS $\mathcal{R}$ is layered wrt. $\mathfrak{L}$:

- L_1 Every term in $\mathcal{T}(\mathcal{F},\mathcal{V})$ has a non-empty top
- L_2 If $x \in \mathcal{V}$ and $C \in \mathcal{C}$, then $C[x]_p \in \mathfrak{L}$ if and only if $C[\Box]_p \in \mathfrak{L}$
- L_3 If $L, N \in \mathfrak{L}$, $p \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(L)$, and $L|_p \sqcup N$ is defined, then $L[L|_p \sqcup N]_p \in \mathfrak{L}$
- W If M is the max-top of s, $p \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(M)$, and $s \to_{p,\ell\to r} t$ with $\ell \to r \in \mathcal{R}$, then $M \to_{p,\ell\to r} L$ for some $L \in \mathfrak{L}$
- C_1 In (W), either L is the max-top of t or $L = \Box$
- C_2 If $L, N \in \mathfrak{L}$ and $L \sqsubseteq N$, then $L[N|_p]_p \in \mathfrak{L}$ for any $p \in \mathcal{P}\mathrm{os}_{\Box}(L)$

- (W) respectively (C_1) are most difficult to formalize

Table of Contents

- Introduction
- Layer Systems
- Formalization

Challenges

Modular proof setup

- separate main results and applications
- avoid boilerplate

Challenges

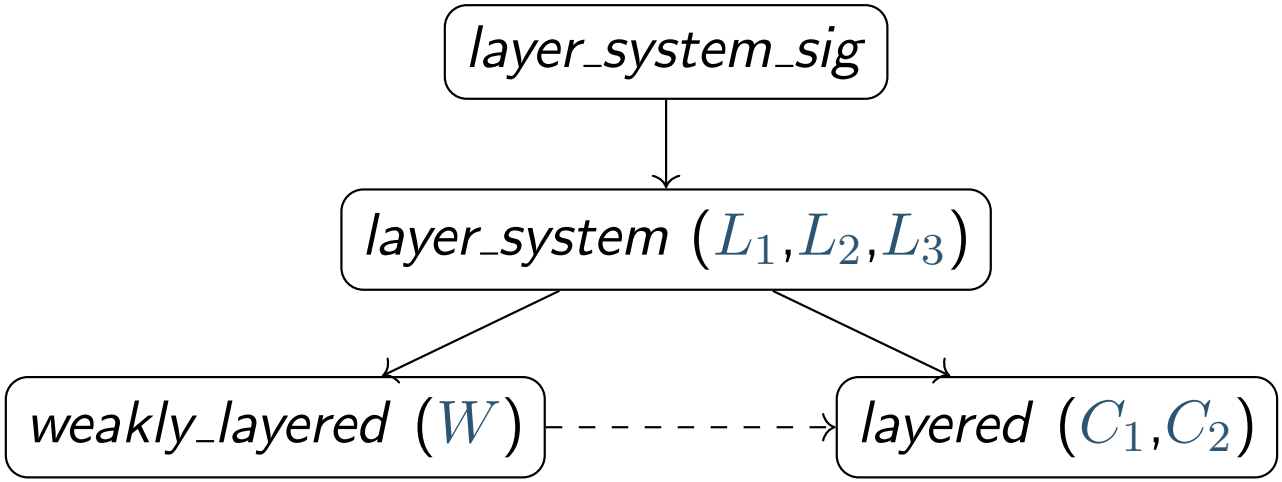
Modular proof setup

- separate main results and applications
- avoid boilerplate

Intuition

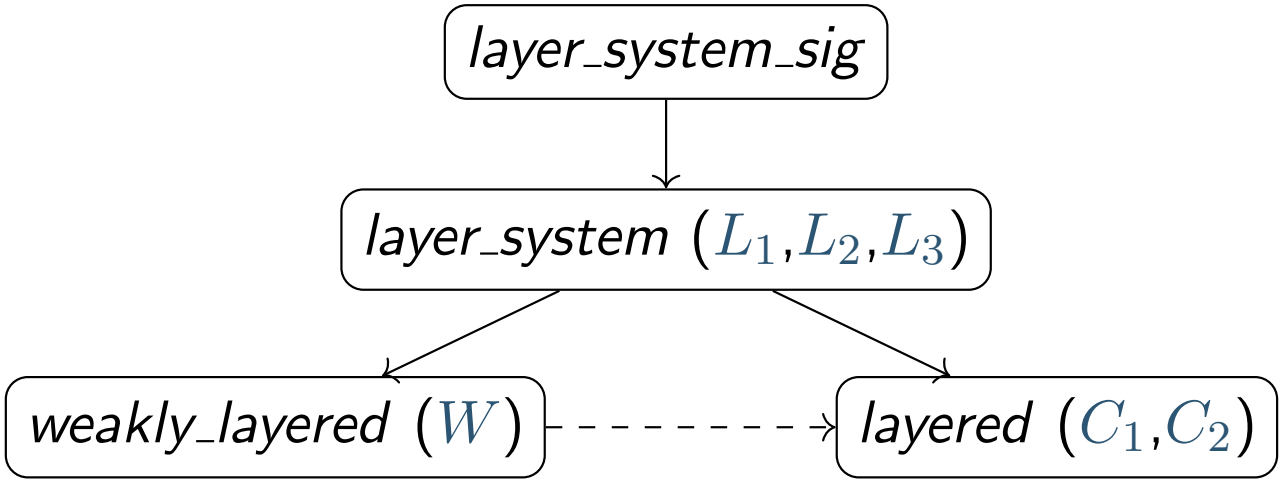
- intuition, and pictures of terms, easily convince humans
- for formalization, algebraic properties are often required
- catch phrases: *"it's easy to see ..."*, *"obviously, ..."*

Isabelle locales



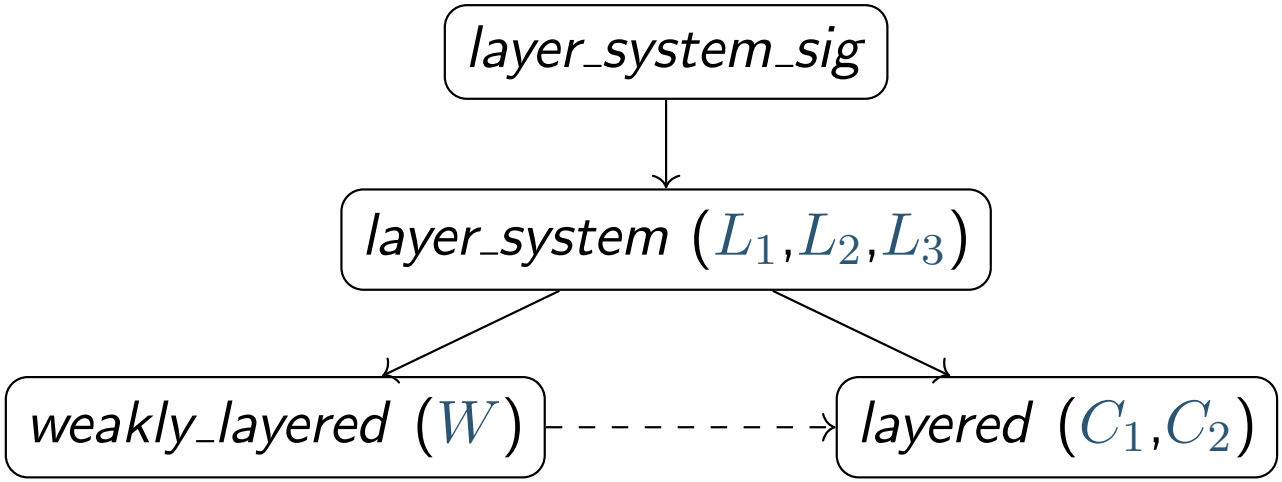
- locales bundle assumptions and conclusions
- locales can be instantiated (used for applications)

Isabelle locales



- locales bundle assumptions and conclusions
- locales can be instantiated (used for applications)
- avoid repetition
- define interfaces for formalization efforts

Isabelle locales

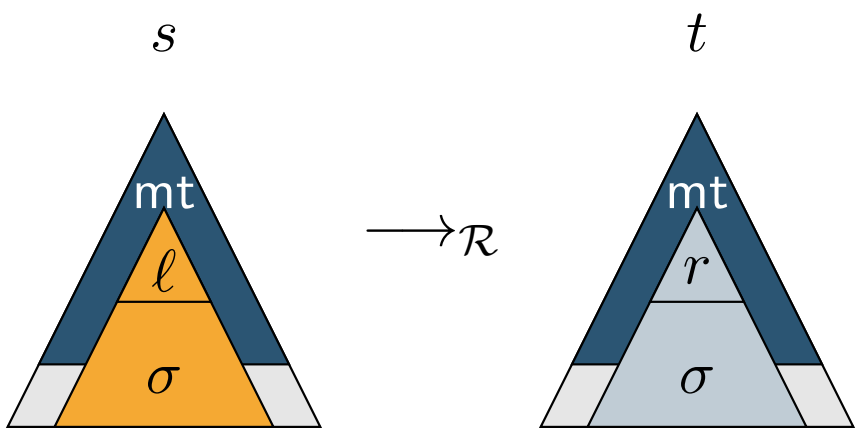


- locales bundle assumptions and conclusions
- locales can be instantiated (used for applications)
- avoid repetition
- define interfaces for formalization efforts

$\Rightarrow$ solves modularization challenge

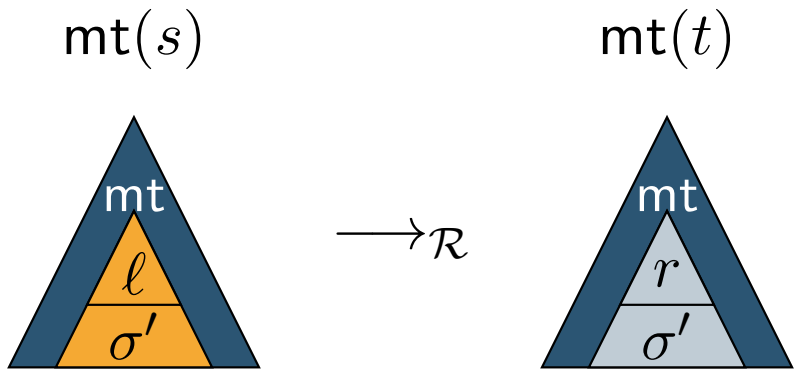
Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top



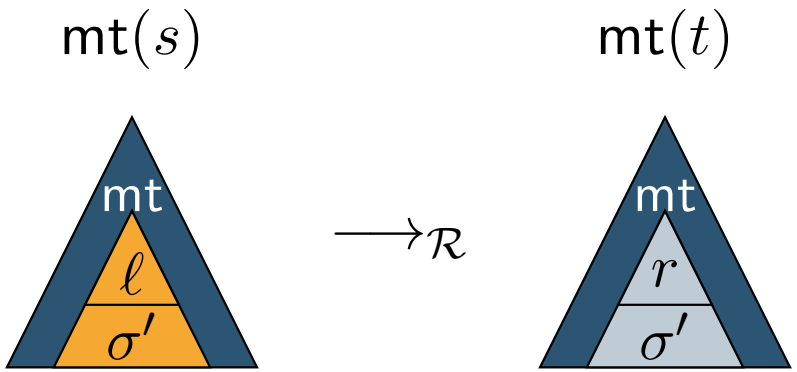
Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top



Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top

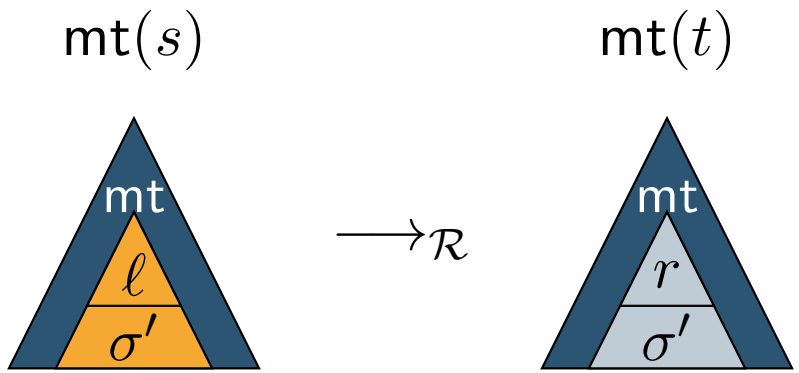


- algebraic computation:

$$\mathsf{mt}(s)$$

Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top



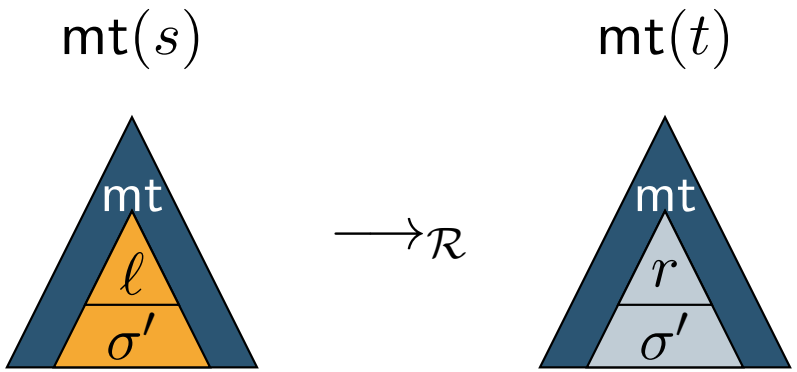
- algebraic computation:

$$\mathsf{mt}(s) = \mathsf{mt}'(C[\ell \cdot \sigma])$$

assuming $\mathsf{mt}(s) = \mathsf{mt}'(s)$, where mt' computes the maxtop with respect to a subset of the layers

Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top



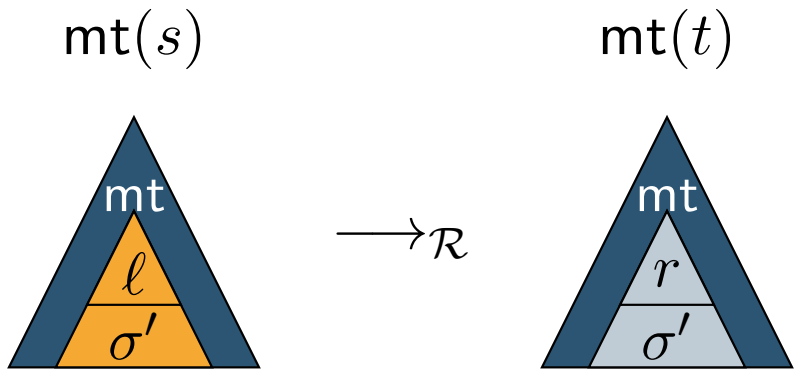
- algebraic computation:

$$\mathsf{mt}(s) = \mathsf{mt}'(C[\ell \cdot \sigma]) \stackrel{\star}{=} D[\mathsf{mt}'(\ell \cdot \sigma)]$$

- $D = \mathsf{mt}'(C)$

Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top



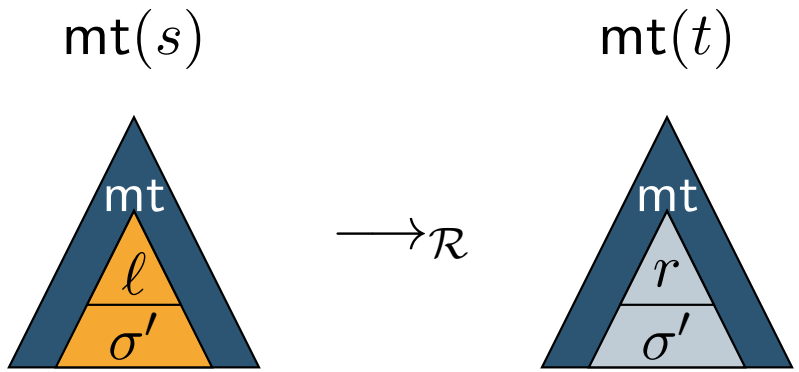
- algebraic computation:

$$\mathsf{mt}(s) = \mathsf{mt}'(C[\ell \cdot \sigma]) \stackrel{\star}{=} D[\mathsf{mt}'(\ell \cdot \sigma)] \stackrel{\dagger}{=} D[\mathsf{mt}'(\ell) \cdot \sigma']$$

- $D = \mathsf{mt}'(C)$, $\sigma' = \mathsf{mt}' \circ \sigma$

Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top



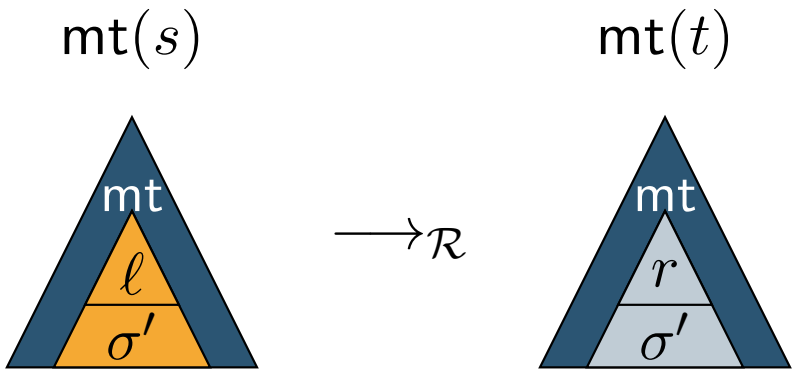
- algebraic computation:

$$\mathsf{mt}(s) = \mathsf{mt}'(C[\ell \cdot \sigma]) \overset{\star}{=} D[\mathsf{mt}'(\ell \cdot \sigma)] \overset{\dagger}{=} D[\mathsf{mt}'(\ell) \cdot \sigma'] \overset{\blacksquare}{=} D[\ell \cdot \sigma']$$

- $D = \mathsf{mt}'(C),\ \sigma' = \mathsf{mt}' \circ \sigma$

Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top



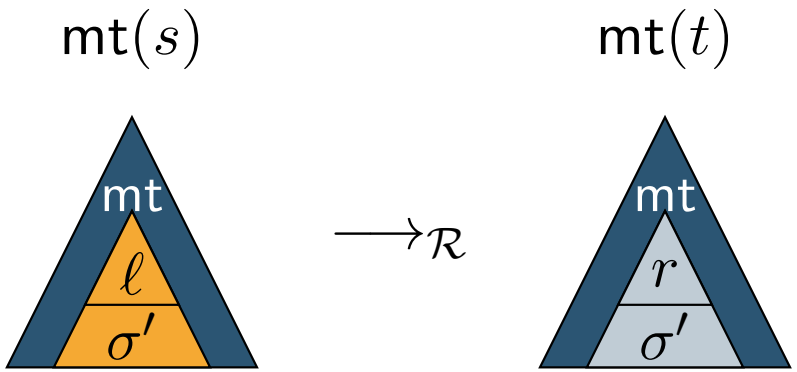
- algebraic computation:

$$\mathsf{mt}(s) = \mathsf{mt}'(C[\ell \cdot \sigma]) \stackrel{\star}{=} D[\mathsf{mt}'(\ell \cdot \sigma)] \stackrel{\dagger}{=} D[\mathsf{mt}'(\ell) \cdot \sigma']$$
$$\stackrel{\blacksquare}{=} D[\ell \cdot \sigma'] \to_{p,\ell\to r} D[r \cdot \sigma']$$

- $D = \mathsf{mt}'(C),\ \sigma' = \mathsf{mt}' \circ \sigma$

Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top



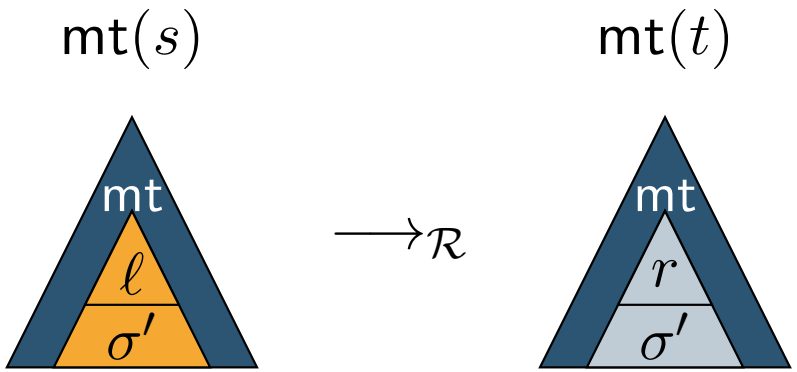
- algebraic computation:

$$\mathsf{mt}(s) = \mathsf{mt}'(C[\ell \cdot \sigma]) \stackrel{\star}{=} D[\mathsf{mt}'(\ell \cdot \sigma)] \stackrel{\dagger}{=} D[\mathsf{mt}'(\ell) \cdot \sigma']$$
$$\stackrel{\blacksquare}{=} D[\ell \cdot \sigma'] \to_{p,\ell \to r} D[r \cdot \sigma'] \stackrel{\blacksquare}{=} D[\mathsf{mt}'(r) \cdot \sigma']$$

- $D = \mathsf{mt}'(C),\ \sigma' = \mathsf{mt}' \circ \sigma$

Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top



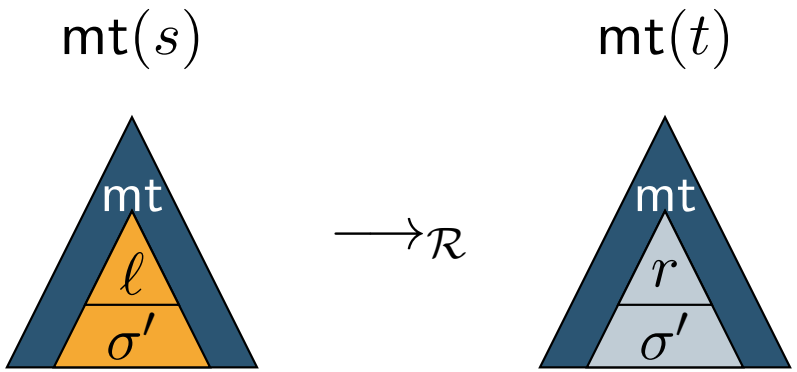
- algebraic computation:

$$\begin{aligned}\mathsf{mt}(s) = \mathsf{mt}'(C[\ell \cdot \sigma]) &\stackrel{\star}{=} D[\mathsf{mt}'(\ell \cdot \sigma)] \stackrel{\dagger}{=} D[\mathsf{mt}'(\ell) \cdot \sigma'] \\ &\stackrel{\blacksquare}{=} D[\ell \cdot \sigma'] \to_{p,\ell\to r} D[r \cdot \sigma'] \stackrel{\blacksquare}{=} D[\mathsf{mt}'(r) \cdot \sigma'] \\ &\stackrel{\dagger}{=} D[\mathsf{mt}'(r \cdot \sigma)]\end{aligned}$$

- $D = \mathsf{mt}'(C)$, $\sigma' = \mathsf{mt}' \circ \sigma$

Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top



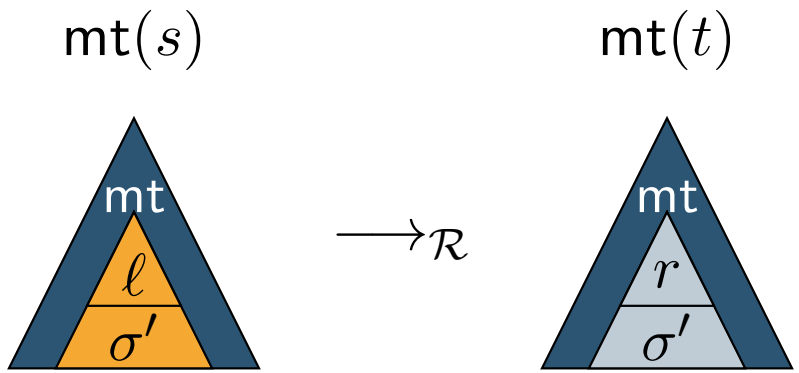
- algebraic computation:

$$\begin{aligned}\mathsf{mt}(s) = \mathsf{mt}'(C[\ell \cdot \sigma]) &\stackrel{\star}{=} D[\mathsf{mt}'(\ell \cdot \sigma)] \stackrel{\dagger}{=} D[\mathsf{mt}'(\ell) \cdot \sigma'] \\ &\stackrel{\blacksquare}{=} D[\ell \cdot \sigma'] \to_{p,\ell\to r} D[r \cdot \sigma'] \stackrel{\blacksquare}{=} D[\mathsf{mt}'(r) \cdot \sigma'] \\ &\stackrel{\dagger}{=} D[\mathsf{mt}'(r \cdot \sigma)] \stackrel{\star}{=} \mathsf{mt}'(C[r \cdot \sigma])\end{aligned}$$

- $D = \mathsf{mt}'(C),\ \sigma' = \mathsf{mt}' \circ \sigma$

Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top



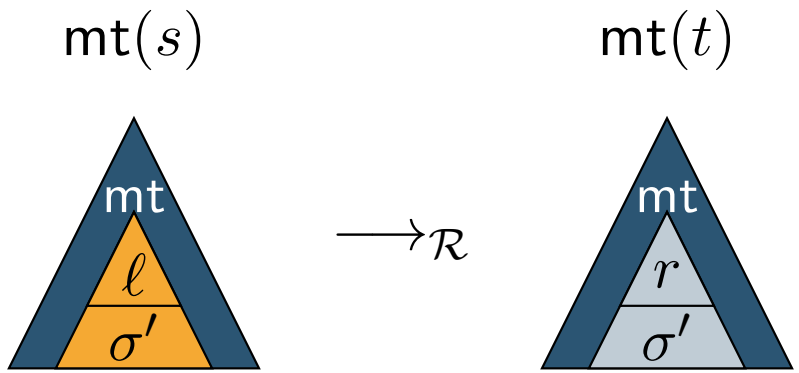
- algebraic computation:

$$\begin{aligned}\mathsf{mt}(s) = \mathsf{mt}'(C[\ell \cdot \sigma]) &\stackrel{\star}{=} D[\mathsf{mt}'(\ell \cdot \sigma)] \stackrel{\dagger}{=} D[\mathsf{mt}'(\ell) \cdot \sigma'] \\ &\stackrel{\blacksquare}{=} D[\ell \cdot \sigma'] \to_{p,\ell\to r} D[r \cdot \sigma'] \stackrel{\blacksquare}{=} D[\mathsf{mt}'(r) \cdot \sigma'] \\ &\stackrel{\dagger}{=} D[\mathsf{mt}'(r \cdot \sigma)] \stackrel{\star}{=} \mathsf{mt}'(C[r \cdot \sigma]) = \mathsf{mt}'(t)\end{aligned}$$

- $D = \mathsf{mt}'(C),\ \sigma' = \mathsf{mt}' \circ \sigma$

Example of algebraization

Idea for C_1: (outer) steps can be simulated in max-top



- algebraic computation:

$$\begin{aligned}\mathsf{mt}(s) = \mathsf{mt}'(C[\ell\cdot\sigma]) &\stackrel{\star}{=} D[\mathsf{mt}'(\ell\cdot\sigma)] \stackrel{\dagger}{=} D[\mathsf{mt}'(\ell)\cdot\sigma'] \\ &\stackrel{\blacksquare}{=} D[\ell\cdot\sigma'] \to_{p,\ell\to r} D[r\cdot\sigma'] \stackrel{\blacksquare}{=} D[\mathsf{mt}'(r)\cdot\sigma'] \\ &\stackrel{\dagger}{=} D[\mathsf{mt}'(r\cdot\sigma)] \stackrel{\star}{=} \mathsf{mt}'(C[r\cdot\sigma]) = \mathsf{mt}'(t)\end{aligned}$$

- $D = \mathsf{mt}'(C),\ \sigma' = \mathsf{mt}' \circ \sigma$

either $\mathsf{mt}'(t) = \mathsf{mt}(t)$, or $\mathsf{mt}'(t) = \Box$

Progress

- general setup (definitions, locales), many technical lemmas — done
- proofs of the main results — in progress

Progress

- general setup (definitions, locales), many technical lemmas — done
- proofs of the main results — in progress
- applications
  - modularity — done
  - many-sorted persistence — done
  - order-sorted persistence — tbd
  - currying — done

Progress

- general setup (definitions, locales), many technical lemmas — done
- proofs of the main results — in progress
- applications
  - ◇ modularity — done
  - ◇ many-sorted persistence — done
  - ◇ order-sorted persistence — tbd
  - ◇ currying — done
  - ◇ layer preservation — future work
  - ◇ quasi-ground systems — future work

Conclusion

Summary

- effort to formalize persistence, currying, and related results
- using layer systems
- separated applications from main results
- 10k lines of Isabelle so far
- technically, no result yet

Conclusion

Summary

- effort to formalize persistence, currying, and related results
- using layer systems
- separated applications from main results
- 10k lines of Isabelle so far
- technically, no result yet

Open tasks

- finish main results
- order-sorted persistence
- further applications

Conclusion

Summary

- effort to formalize persistence, currying, and related results
- using layer systems
- separated applications from main results
- 10k lines of Isabelle so far
- technically, no result yet

Open tasks

- finish main results
- order-sorted persistence
- further applications

Thanks!