Modular Tableaux Calculi for Bunched Logics and Separation Theories

BCTCS 2018

Simon Docherty

University College London

Tuesday 27th March 2018

Joint work with David Pym[1]

[1] S. Docherty and D. Pym. Modular Tableaux Calculi for Separation Theories. *FoSSaCS '18*

What is this work about?

- Micro Level: systematic definition of proof systems for the program verification formalism Separation Logic.

What is this work about?

- Micro Level: systematic definition of proof systems for the program verification formalism Separation Logic.
- Macro Level: technique for overcoming expressivity/definability trade-off in logic.

Bunched Logics

What Are Bunched Logics?

- Classical/intuitionistic propositional logic extended with substructural connectives lacking contraction[2]:

$$\varphi * \varphi \not\equiv \varphi.$$

[2] P. O'Hearn and D. Pym. The Logic of Bunched Implications. *Bulletin of Symbolic Logic*, 1999.

[3] S. Ishtiaq and P. O'Hearn. BI As An Assertion Language for Mutable Data Structures. LICS '01.

What Are Bunched Logics?

- Classical/intuitionistic propositional logic extended with substructural connectives lacking contraction[2]:

$$\varphi * \varphi \not\equiv \varphi.$$

- Proof theory: contexts are trees (bunches), not lists.

[2] P. O'Hearn and D. Pym. The Logic of Bunched Implications. *Bulletin of Symbolic Logic*, 1999.

[3] S. Ishtiaq and P. O'Hearn. BI As An Assertion Language for Mutable Data Structures. LICS '01.

What Are Bunched Logics?

- Classical/intuitionistic propositional logic extended with substructural connectives lacking contraction[2]:

$$\varphi * \varphi \not\equiv \varphi.$$

- Proof theory: contexts are trees (bunches), not lists.
- Formulae describe composable/comparable resources.

[2] P. O'Hearn and D. Pym. The Logic of Bunched Implications. *Bulletin of Symbolic Logic*, 1999.

[3] S. Ishtiaq and P. O'Hearn. BI As An Assertion Language for Mutable Data Structures. LICS '01.

What Are Bunched Logics?

- Classical/intuitionistic propositional logic extended with substructural connectives lacking contraction[2]:

$$\varphi * \varphi \not\equiv \varphi.$$

- Proof theory: contexts are trees (bunches), not lists.
- Formulae describe composable/comparable resources.
- Applications across computer science: program verification[3], security modelling, resource reasoning.

[2] P. O'Hearn and D. Pym. The Logic of Bunched Implications. *Bulletin of Symbolic Logic*, 1999.

[3] S. Ishtiaq and P. O'Hearn. BI As An Assertion Language for Mutable Data Structures. LICS '01.

The Logic of Bunched Implications

Let p range over set of propositional atoms Prop.

$$\varphi ::= \mathrm{p} \mid \top \mid \bot \mid \mathrm{I} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi * \varphi \mid \varphi \mathrel{-\!\!*} \varphi$$

[4] D. Galmiche, D. Méry and D. Pym. The Semantics of BI and Resource Tableaux. *Math. Struc. Comp. Sci.*, 2005.

[5] J. Brotherston and M. Kanovich. Undecidability of Propositional Separation Logic and Its Neighbours. *J. ACM*, 2014.

The Logic of Bunched Implications

Let p range over set of propositional atoms Prop.

$$\varphi ::= \mathrm{p} \mid \top \mid \bot \mid \mathrm{I} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi * \varphi \mid \varphi \mathrel{-\!\!*} \varphi$$

- BI: standard connectives = intuitionistic logic; decidable[4]

[4] D. Galmiche, D. Méry and D. Pym. The Semantics of BI and Resource Tableaux. *Math. Struc. Comp. Sci.*, 2005.

[5] J. Brotherston and M. Kanovich. Undecidability of Propositional Separation Logic and Its Neighbours. *J. ACM*, 2014.

The Logic of Bunched Implications

Let p range over set of propositional atoms Prop.

$$\varphi ::= \mathrm{p} \mid \top \mid \bot \mid \mathrm{I} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi * \varphi \mid \varphi \mathrel{-\!\!*} \varphi$$

- BI: standard connectives = intuitionistic logic; decidable[4]
- Boolean BI (BBI): standard connectives = classical logic; undecidable[5]

[4] D. Galmiche, D. Méry and D. Pym. The Semantics of BI and Resource Tableaux. *Math. Struc. Comp. Sci.*, 2005.

[5] J. Brotherston and M. Kanovich. Undecidability of Propositional Separation Logic and Its Neighbours. *J. ACM*, 2014.

The Logic of Bunched Implications

Let p range over set of propositional atoms Prop.

$$\varphi ::= \mathrm{p} \mid \top \mid \bot \mid \mathrm{I} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi * \varphi \mid \varphi \mathrel{-\!\!*} \varphi$$

- BI: standard connectives = intuitionistic logic; decidable[4]
- Boolean BI (BBI): standard connectives = classical logic; undecidable[5]
- $*$ and $\mathrel{-\!\!*}$ adjoint: $(\varphi * \psi) \rightarrow \chi \equiv \varphi \rightarrow (\psi \mathrel{-\!\!*} \chi)$;
- $*$ associative: $(\varphi * \psi) * \chi \equiv \varphi * (\psi * \chi)$;
- $*$ commutative: $\varphi * \psi \equiv \psi * \varphi$;
- I is unit for $*$: $\varphi * \mathrm{I} \equiv \varphi$;

[4] D. Galmiche, D. Méry and D. Pym. The Semantics of BI and Resource Tableaux. *Math. Struc. Comp. Sci.*, 2005.

[5] J. Brotherston and M. Kanovich. Undecidability of Propositional Separation Logic and Its Neighbours. *J. ACM*, 2014.

Resource Semantics

Definition (**BI Model**)

- Set of resources X;

Resource Semantics

Definition (**BI Model**)

- Set of resources X;
- Composition $\circ : X^2 \to \mathcal{P}(X)$, commutative and associative;

Resource Semantics

Definition (**BI Model**)

- Set of resources X;
- Composition $\circ : X^2 \to \mathcal{P}(X)$, commutative and associative;
- Set of unit resources E;

Resource Semantics

Definition (**BI Model**)

- Set of resources X;
- Composition $\circ : X^2 \to \mathcal{P}(X)$, commutative and associative;
- Set of unit resources E;
- Partial order $\leq$ for comparing resources, compatible with $\circ$;

Resource Semantics

Definition (**BI Model**)

- Set of resources X;
- Composition $\circ : X^2 \to \mathcal{P}(X)$, commutative and associative;
- Set of unit resources E;
- Partial order $\leq$ for comparing resources, compatible with $\circ$;
- Monotonic valuation $\mathcal{V} : \mathrm{Prop} \to \mathcal{P}(X)$.

Resource Semantics

Definition (**BI Model**)

- Set of resources X;
- Composition $\circ : X^2 \to \mathcal{P}(X)$, commutative and associative;
- Set of unit resources E;
- Partial order $\leq$ for comparing resources, compatible with $\circ$;
- Monotonic valuation $\mathcal{V} : \mathrm{Prop} \to \mathcal{P}(X)$.

Eg: $(\mathbb{N}, +, \leq, \{0\})$.

Resource Semantics

Definition (**BI Model**)

- Set of resources X;
- Composition $\circ : X^2 \to \mathcal{P}(X)$, commutative and associative;
- Set of unit resources E;
- Partial order $\leq$ for comparing resources, compatible with $\circ$;
- Monotonic valuation $\mathcal{V} : \mathrm{Prop} \to \mathcal{P}(X)$.

Eg: $(\mathbb{N}, +, \leq, \{0\})$.

- $x \vDash \mathrm{I}$ iff $x \in E$.
- $x \vDash \varphi * \psi$ iff there exist $\exists x', y, z$ s.t. $x \geq x' \in y \circ z, y \vDash \varphi, z \vDash \psi$;

Resource Semantics

Definition (**BI Model**)

- Set of resources X;
- Composition $\circ : X^2 \to \mathcal{P}(X)$, commutative and associative;
- Set of unit resources E;
- Partial order $\leq$ for comparing resources, compatible with $\circ$;
- Monotonic valuation $\mathcal{V} : \mathrm{Prop} \to \mathcal{P}(X)$.

Eg: $(\mathbb{N}, +, \leq, \{0\})$.

- $x \vDash \mathrm{I}$ iff $x \in E$.
- $x \vDash \varphi * \psi$ iff there exist $\exists x', y, z$ s.t. $x \geq x' \in y \circ z, y \vDash \varphi, z \vDash \psi$;

For **BBI**: $\leq$ is $=$.

Separation Theories

Separation Logic

- Verification formalism for programs that mutate shared data structures[6].

[6] J. Reynolds. Separation Logic: A Logic for Shared Mutable Data Structures. *LICS 2002*.

Separation Logic

- Verification formalism for programs that mutate shared data structures[6].
- RAM model: heaps (memory allocations), $\circ$ composes disjoint heaps, E is empty heap, $\leq$ is heap extension.

[6] J. Reynolds. Separation Logic: A Logic for Shared Mutable Data Structures. *LICS 2002*.

Separation Logic

- Verification formalism for programs that mutate shared data structures[6].
- RAM model: heaps (memory allocations), $\circ$ composes disjoint heaps, E is empty heap, $\leq$ is heap extension.
- Calculus of Hoare triples $\{\varphi\}C\{\psi\}$: if memory in state φ and C executes, memory will be in state ψ afterwards.

[6] J. Reynolds. Separation Logic: A Logic for Shared Mutable Data Structures. *LICS 2002*.

Separation Logic

- Verification formalism for programs that mutate shared data structures[6].
- RAM model: heaps (memory allocations), $\circ$ composes disjoint heaps, E is empty heap, $\leq$ is heap extension.
- Calculus of Hoare triples $\{\varphi\}C\{\psi\}$: if memory in state φ and C executes, memory will be in state ψ afterwards.
- Frame rule $\Rightarrow$ scalability:

$$\frac{\{\varphi\}\,C\,\{\psi\}}{\{\varphi * \chi\}\,C\,\{\psi * \chi\}},$$

where C doesn't modify memory described by χ.

[6] J. Reynolds. Separation Logic: A Logic for Shared Mutable Data Structures. *LICS 2002*.

Separation Logic

- Verification formalism for programs that mutate shared data structures[6].
- RAM model: heaps (memory allocations), $\circ$ composes disjoint heaps, E is empty heap, $\leq$ is heap extension.
- Calculus of Hoare triples $\{\varphi\}C\{\psi\}$: if memory in state φ and C executes, memory will be in state ψ afterwards.
- Frame rule $\Rightarrow$ scalability:

$$\frac{\{\varphi\}\,C\,\{\psi\}}{\{\varphi * \chi\}\,C\,\{\psi * \chi\}},$$

  where C doesn't modify memory described by χ.
- Different memory models for different tasks $\Rightarrow$ zoo of separation logics.

[6] J. Reynolds. Separation Logic: A Logic for Shared Mutable Data Structures. *LICS 2002*.

A Taxonomy of Separation Logics: Separation Theories

Abstraction of separation logics as **Separation Algebras**[7 8 9]: (B)BI models in which $\leq$, $\circ$, E satisfy additional first-order **Separation Theories**.

[7] C. Calcagno et al. Local Action and Abstract Separation Logic. *LICS 2007*.
[8] R. Dockins et al. A Fresh Look at Separation Algebras... *APLAS 2009*.
[9] Q. Cao et al. Bringing Order To The Separation Logic Jungle. *APLAS 2017*.

A Taxonomy of Separation Logics: Separation Theories

Abstraction of separation logics as **Separation Algebras**[7 8 9]: (B)BI models in which $\leq$, $\circ$, E satisfy additional first-order **Separation Theories**. For e.g.:

Partial Determinism	$\forall x, y, z, z'(z \in x \circ y \wedge z' \in x \circ y \rightarrow z = z')$
Total	$\forall x, y(\exists z(z \in x \circ y))$

A Taxonomy of Separation Logics: Separation Theories

Abstraction of separation logics as **Separation Algebras**[7 8 9]: (B)BI models in which $\leq$, $\circ$, E satisfy additional first-order **Separation Theories**. For e.g.:

Partial Determinism	$\forall x, y, z, z'(z \in x \circ y \wedge z' \in x \circ y \rightarrow z = z')$
Total	$\forall x, y(\exists z(z \in x \circ y))$
Single Unit	$\forall x, x'(x \in E \wedge x' \in E \rightarrow x = x')$
Disjointness	$\forall x, y(x \in y \circ y \rightarrow y \in E)$
Unit Self-joining	$\forall x(x \in E \rightarrow x \in x \circ x)$

A Taxonomy of Separation Logics: Separation Theories

Abstraction of separation logics as **Separation Algebras**[7 8 9]: (B)BI models in which $\leq$, $\circ$, E satisfy additional first-order **Separation Theories**. For e.g.:

Partial Determinism	$\forall x, y, z, z'(z \in x \circ y \wedge z' \in x \circ y \rightarrow z = z')$
Total	$\forall x, y(\exists z(z \in x \circ y))$
Single Unit	$\forall x, x'(x \in E \wedge x' \in E \rightarrow x = x')$
Disjointness	$\forall x, y(x \in y \circ y \rightarrow y \in E)$
Unit Self-joining	$\forall x(x \in E \rightarrow x \in x \circ x)$
Non-branching	$\forall x, y, y'(x \leq y \wedge x \leq y' \rightarrow y \leq y' \vee y' \leq y)$
Increasing	$\forall x, y, z(z \in x \circ y \rightarrow y \leq z)$

[7] C. Calcagno et al. Local Action and Abstract Separation Logic. *LICS 2007*.
[8] R. Dockins et al. A Fresh Look at Separation Algebras... *APLAS 2009*.
[9] Q. Cao et al. Bringing Order To The Separation Logic Jungle. *APLAS 2017*.

Some Separation Theories Are Undefinable in (B)BI

Given logic $\mathcal{L}$, a model property P is **$\mathcal{L}$-definable** if there exists an $\mathcal{L}$ – *formula* φ such that

$$\varphi \text{ valid in } M \text{ iff } M \text{ is a } P\text{-model.}$$

[10] J. Brotherston and J. Villard. Parametric Completeness for Separation Theories. *POPL 2014*.

Some Separation Theories Are Undefinable in (B)BI

Given logic $\mathcal{L}$, a model property P is **$\mathcal{L}$-definable** if there exists an $\mathcal{L}$ – *formula* φ such that

$$\varphi \text{ valid in } M \text{ iff } M \text{ is a } P\text{-model.}$$

Theorem (Brotherston and Villard)

Partial Determinism, Cancellativity, Single Unit and Disjointness are not BBI-definable[10].

[10] J. Brotherston and J. Villard. Parametric Completeness for Separation Theories. *POPL 2014*.

Some Separation Theories Are Undefinable in (B)BI

Given logic $\mathcal{L}$, a model property P is **$\mathcal{L}$-definable** if there exists an $\mathcal{L}$ – *formula* φ such that

$$\varphi \text{ valid in } M \text{ iff } M \text{ is a } P\text{-model.}$$

Theorem (Brotherston and Villard)

Partial Determinism, Cancellativity, Single Unit and Disjointness are not BBI-definable[10].

[10] J. Brotherston and J. Villard. Parametric Completeness for Separation Theories. *POPL 2014*.

Some Separation Theories Determine Distinct Logics

Theorem (Larchey-Wendling & Galmiche)

BBI models, BBI + Partial Determinism models and BBI + Total models all determine distinct sets of valid formulae[11].

[11] D. Larchey-Wendling and D. Galmiche. The Undecidability of Boolean BI Through Phase Semantics. *LICS 2010*.

Some Separation Theories Determine Distinct Logics

Theorem (Larchey-Wendling & Galmiche)

BBI models, BBI + Partial Determinism models and BBI + Total models all determine distinct sets of valid formulae[11].

Proof Sketch.

$(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \rightarrow \mathrm{I}$ is valid for all Total models, but there exists a Partial model for which it is not. □

[11] D. Larchey-Wendling and D. Galmiche. The Undecidability of Boolean BI Through Phase Semantics. *LICS 2010*.

Some Separation Theories Determine Distinct Logics

Theorem (Larchey-Wendling & Galmiche)

BBI models, BBI + Partial Determinism models and BBI + Total models all determine distinct sets of valid formulae[11].

Proof Sketch.

$(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \to \mathrm{I}$ is valid for all Total models, but there exists a Partial model for which it is not. □

Note: $(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \to \mathrm{I}$ valid in some Partial models so this does not define Total.

[11] D. Larchey-Wendling and D. Galmiche. The Undecidability of Boolean BI Through Phase Semantics. *LICS 2010*.

The Problem

- Separation logics are determined by separation theories with distinct sets of valid formulae.

[12] J. Brotherston and J. Villard. Parametric Completeness for Separation Theories. *POPL '14*.

[13] Z. Hoú, R. Clouston, A. Tiu and R. Goré. Proof Search for Propositional Abstract Separation Logic with Labelled Sequents. *POPL '14*.

The Problem

- Separation logics are determined by separation theories with distinct sets of valid formulae.
- Undefinability ⇒ can't just add (B)BI axioms to standard proof systems to capture separation theories.

[12] J. Brotherston and J. Villard. Parametric Completeness for Separation Theories. *POPL '14*.

[13] Z. Hoú, R. Clouston, A. Tiu and R. Goré. Proof Search for Propositional Abstract Separation Logic with Labelled Sequents. *POPL '14*.

The Problem

- Separation logics are determined by separation theories with distinct sets of valid formulae.
- Undefinability $\Rightarrow$ can't just add (B)BI axioms to standard proof systems to capture separation theories.
- Existing solutions change underlying logic[12] and/or restricted in application[13].

[12] J. Brotherston and J. Villard. Parametric Completeness for Separation Theories. *POPL '14*.

[13] Z. Hoú, R. Clouston, A. Tiu and R. Goré. Proof Search for Propositional Abstract Separation Logic with Labelled Sequents. *POPL '14*.

The Problem

- Separation logics are determined by separation theories with distinct sets of valid formulae.
- Undefinability $\Rightarrow$ can't just add (B)BI axioms to standard proof systems to capture separation theories.
- Existing solutions change underlying logic[12] and/or restricted in application[13].
- **We give a modular tableaux system capturing validity in all (B)BI + Σ models, for any separation theory Σ.**

[12] J. Brotherston and J. Villard. Parametric Completeness for Separation Theories. *POPL '14*.

[13] Z. Hoú, R. Clouston, A. Tiu and R. Goré. Proof Search for Propositional Abstract Separation Logic with Labelled Sequents. *POPL '14*.

Tableaux Calculi for Separation Theories

Basic Units: Labelled Formulae and Constraints

Two basic syntactic entities:

Basic Units: Labelled Formulae and Constraints

Two basic syntactic entities:

1. **Labelled formulae** encoding satisfaction

Basic Units: Labelled Formulae and Constraints

Two basic syntactic entities:

1. **Labelled formulae** encoding satisfaction
   - Sign: $\mathbb{S} \in \{\mathbb{T}, \mathbb{F}\}$ (true/false);

Basic Units: Labelled Formulae and Constraints

Two basic syntactic entities:

1. **Labelled formulae** encoding satisfaction
   - Sign: $\mathbb{S} \in \{\mathbb{T}, \mathbb{F}\}$ (true/false);
   - Label: $x \in \{c_i \mid i \in \mathbb{N}\}$ (representing resources);

Basic Units: Labelled Formulae and Constraints

Two basic syntactic entities:

1. **Labelled formulae** encoding satisfaction
   - Sign: $\mathbb{S} \in \{\mathbb{T}, \mathbb{F}\}$ (true/false);
   - Label: $x \in \{c_i \mid i \in \mathbb{N}\}$ (representing resources);
   - $\mathbb{S}\varphi : x \quad \Rightarrow \quad$ "φ is true/false at the resource x."

Basic Units: Labelled Formulae and Constraints

Two basic syntactic entities:

1. **Labelled formulae** encoding satisfaction
   - Sign: $\mathbb{S} \in \{\mathbb{T}, \mathbb{F}\}$ (true/false);
   - Label: $x \in \{c_i \mid i \in \mathbb{N}\}$ (representing resources);
   - $\mathbb{S}\varphi : x \quad \Rightarrow \quad$ "φ is true/false at the resource x."
2. **Constraints** on labels encoding a (partial) model

Basic Units: Labelled Formulae and Constraints

Two basic syntactic entities:

1. **Labelled formulae** encoding satisfaction
   - Sign: $\mathbb{S} \in \{\mathbb{T}, \mathbb{F}\}$ (true/false);
   - Label: $x \in \{c_i \mid i \in \mathbb{N}\}$ (representing resources);
   - $\mathbb{S}\varphi : x \quad \Rightarrow \quad$ "φ is true/false at the resource x."
2. **Constraints** on labels encoding a (partial) model
   - $x \sim y$: resource $x \leq$ resource y;

Basic Units: Labelled Formulae and Constraints

Two basic syntactic entities:

1. **Labelled formulae** encoding satisfaction
   - Sign: $\mathbb{S} \in \{\mathbb{T}, \mathbb{F}\}$ (true/false);
   - Label: $x \in \{c_i \mid i \in \mathbb{N}\}$ (representing resources);
   - $\mathbb{S}\varphi : x \quad \Rightarrow \quad$ "φ is true/false at the resource x."
2. **Constraints** on labels encoding a (partial) model
   - $x \sim y$: resource $x \leq$ resource y;
   - $R_* yzx$: resource $x \in$ resource $y \circ$ resource z;

Basic Units: Labelled Formulae and Constraints

Two basic syntactic entities:

1. **Labelled formulae** encoding satisfaction
   - Sign: $\mathbb{S} \in \{\mathbb{T}, \mathbb{F}\}$ (true/false);
   - Label: $x \in \{c_i \mid i \in \mathbb{N}\}$ (representing resources);
   - $\mathbb{S}\varphi : x \quad \Rightarrow \quad$ "φ is true/false at the resource x."
2. **Constraints** on labels encoding a (partial) model
   - $x \sim y$: resource $x \leq$ resource y;
   - $R_* yzx$: resource $x \in$ resource $y \circ$ resource z;
   - Ex: resource x is a unit.

How Our Proof Systems Work: Rules, Tableaux, Proofs

- Tableau = tree with nodes given by finite sets of labelled formulae and constraints[14] [15]

[14] D. Galmiche, D. Méry and D. Pym. The Semantics of BI and Resource Tableaux. *Math. Struc. Comp. Sci.*, 2005.

[15] D. Larchey-Wendling. The Formal Strong Completeness of Partial Monoidal BBI. *J. Log. Comp.*, 2014.

How Our Proof Systems Work: Rules, Tableaux, Proofs

- Tableau = tree with nodes given by finite sets of labelled formulae and constraints[14] [15]
- Root = $\{\mathbb{F}\varphi : c_0\}$ where φ is formula trying to prove.

[14] D. Galmiche, D. Méry and D. Pym. The Semantics of BI and Resource Tableaux. *Math. Struc. Comp. Sci.*, 2005.

[15] D. Larchey-Wendling. The Formal Strong Completeness of Partial Monoidal BBI. *J. Log. Comp.*, 2014.

How Our Proof Systems Work: Rules, Tableaux, Proofs

- Tableau = tree with nodes given by finite sets of labelled formulae and constraints[14] [15]
- Root = $\{\mathbb{F}\varphi : c_0\}$ where φ is formula trying to prove.
- Constructed by rules

$$\frac{\mathbb{S}\psi : x \in \mathbb{F} \text{ and } \gamma_1, \ldots, \gamma_n \in C}{\langle \mathcal{F}_1, C_1 \rangle \mid \ldots \mid \langle \mathcal{F}_k, C_k \rangle}$$

[14] D. Galmiche, D. Méry and D. Pym. The Semantics of BI and Resource Tableaux. *Math. Struc. Comp. Sci.*, 2005.

[15] D. Larchey-Wendling. The Formal Strong Completeness of Partial Monoidal BBI. *J. Log. Comp.*, 2014.

How Our Proof Systems Work: Rules, Tableaux, Proofs

- Tableau = tree with nodes given by finite sets of labelled formulae and constraints[14] [15]
- Root = $\{\mathbb{F}\varphi : c_0\}$ where φ is formula trying to prove.
- Constructed by rules

$$\frac{\mathbb{S}\psi : x \in \mathbb{F} \text{ and } \gamma_1, \ldots, \gamma_n \in \mathcal{C}}{\langle \mathcal{F}_1, \mathcal{C}_1 \rangle \mid \ldots \mid \langle \mathcal{F}_k, \mathcal{C}_k \rangle}$$

"If $\mathbb{S}\psi : x$ and constraints γ_i on branch, branch extends into k branches through addition of formulae/constraints $\langle \mathcal{F}_i, \mathcal{C}_i \rangle$."

[14] D. Galmiche, D. Méry and D. Pym. The Semantics of BI and Resource Tableaux. *Math. Struc. Comp. Sci.*, 2005.

[15] D. Larchey-Wendling. The Formal Strong Completeness of Partial Monoidal BBI. *J. Log. Comp.*, 2014.

How Our Proof Systems Work: Rules, Tableaux, Proofs

- Tableau = tree with nodes given by finite sets of labelled formulae and constraints[14] [15]
- Root = $\{\mathbb{F}\varphi : c_0\}$ where φ is formula trying to prove.
- Constructed by rules

$$\frac{\mathbb{S}\psi : x \in \mathbb{F} \text{ and } \gamma_1, \ldots, \gamma_n \in \mathcal{C}}{\langle \mathcal{F}_1, \mathcal{C}_1 \rangle \mid \ldots \mid \langle \mathcal{F}_k, \mathcal{C}_k \rangle}$$

  "If $\mathbb{S}\psi : x$ and constraints γ_i on branch, branch extends into k branches through addition of formulae/constraints $\langle \mathcal{F}_i, \mathcal{C}_i \rangle$."

- Closed branch: contains contradictory formulae/constraints. Proof: all branches closed.

[14] D. Galmiche, D. Méry and D. Pym. The Semantics of BI and Resource Tableaux. *Math. Struc. Comp. Sci.*, 2005.

[15] D. Larchey-Wendling. The Formal Strong Completeness of Partial Monoidal BBI. *J. Log. Comp.*, 2014.

Basic Tableaux Rules for Bunched Logic

- Each connective has $\mathbb{T}$ and $\mathbb{F}$ rules for decomposing formulae.

Basic Tableaux Rules for Bunched Logic

- Each connective has $\mathbb{T}$ and $\mathbb{F}$ rules for decomposing formulae. Examples for BBI:

$$\langle \mathbb{T}* \rangle \ \frac{\mathbb{T}\varphi * \psi : x \in \mathcal{F}}{\langle \{\mathbb{T}\varphi : c_i, \mathbb{T}\psi : c_j\}, \{R_* c_i c_j x\} \rangle} \quad \langle \mathbb{F}* \rangle \ \frac{\mathbb{F}\varphi * \psi : x \in \mathcal{F} \text{ and } R_* yzx \in \mathcal{C}}{\langle \{\mathbb{F}\varphi : y\}, \emptyset \rangle \mid \langle \{\mathbb{F}\psi : z\}, \emptyset \rangle}$$

Basic Tableaux Rules for Bunched Logic

- Each connective has $\mathbb{T}$ and $\mathbb{F}$ rules for decomposing formulae. Examples for BBI:

$$\langle \mathbb{T}* \rangle \ \frac{\mathbb{T}\varphi * \psi : x \in \mathcal{F}}{\langle \{\mathbb{T}\varphi : c_i, \mathbb{T}\psi : c_j\}, \{R_* c_i c_j x\} \rangle} \quad \langle \mathbb{F}* \rangle \ \frac{\mathbb{F}\varphi * \psi : x \in \mathcal{F} \text{ and } R_* yzx \in \mathcal{C}}{\langle \{\mathbb{F}\varphi : y\}, \emptyset \rangle \mid \langle \{\mathbb{F}\psi : z\}, \emptyset \rangle}$$

- Rules acting on constraints.

Basic Tableaux Rules for Bunched Logic

- Each connective has $\mathbb{T}$ and $\mathbb{F}$ rules for decomposing formulae. Examples for BBI:

$$\langle \mathbb{T}* \rangle \quad \frac{\mathbb{T}\varphi * \psi : x \in \mathcal{F}}{\langle \{\mathbb{T}\varphi : c_i, \mathbb{T}\psi : c_j\}, \{R_* c_i c_j x\} \rangle} \qquad \langle \mathbb{F}* \rangle \quad \frac{\mathbb{F}\varphi * \psi : x \in \mathcal{F} \text{ and } R_* yzx \in \mathcal{C}}{\langle \{\mathbb{F}\varphi : y\}, \emptyset \rangle \mid \langle \{\mathbb{F}\psi : z\}, \emptyset \rangle}$$

- Rules acting on constraints. Examples for BBI:

$$\langle \text{Comm} \rangle \quad \frac{R_* xyz \in \mathcal{C}}{\langle \emptyset, \{R_* yxz\} \rangle} \qquad \langle \text{Sym} \rangle \quad \frac{x \sim y \in \mathcal{C}}{\langle \emptyset, \{y \sim x\} \rangle}$$

A Useful Property of Separation Theories

Definition

A **coherent formula** is a first-order formula of the form

$$\forall \vec{x}(A_1(\vec{x}) \wedge \cdots \wedge A_n(\vec{x}) \rightarrow \exists \overrightarrow{y_1} B_1(\vec{x}, \overrightarrow{y_1}) \vee \cdots \vee \exists \overrightarrow{y_m} B_m(\vec{x}, \overrightarrow{y_m}))$$

such that

A Useful Property of Separation Theories

Definition

A **coherent formula** is a first-order formula of the form

$$\forall \vec{x}(A_1(\vec{x}) \wedge \cdots \wedge A_n(\vec{x}) \rightarrow \exists \overrightarrow{y_1} B_1(\vec{x}, \overrightarrow{y_1}) \vee \cdots \vee \exists \overrightarrow{y_m} B_m(\vec{x}, \overrightarrow{y_m}))$$

such that

- Each A_i an atomic first-order formula

A Useful Property of Separation Theories

Definition

A **coherent formula** is a first-order formula of the form

$$\forall \vec{x}(A_1(\vec{x}) \wedge \cdots \wedge A_n(\vec{x}) \rightarrow \exists \overrightarrow{y_1} B_1(\vec{x}, \overrightarrow{y_1}) \vee \cdots \vee \exists \overrightarrow{y_m} B_m(\vec{x}, \overrightarrow{y_m}))$$

such that

- Each A_i an atomic first-order formula
  E.g. $Rxyz$, $y = z$, Ea, $x \leq y$.

A Useful Property of Separation Theories

Definition

A **coherent formula** is a first-order formula of the form

$$\forall \vec{x}(A_1(\vec{x}) \wedge \cdots \wedge A_n(\vec{x}) \rightarrow \exists \overrightarrow{y_1} B_1(\vec{x}, \overrightarrow{y_1}) \vee \cdots \vee \exists \overrightarrow{y_m} B_m(\vec{x}, \overrightarrow{y_m}))$$

such that

- Each A_i an atomic first-order formula
  E.g. $Rxyz$, $y = z$, Ea, $x \leq y$.
- Each B_i a finite conjunction of atomic first-order formulae

A Useful Property of Separation Theories

Definition

A **coherent formula** is a first-order formula of the form

$$\forall \vec{x}(A_1(\vec{x}) \wedge \cdots \wedge A_n(\vec{x}) \rightarrow \exists \overrightarrow{y_1} B_1(\vec{x}, \overrightarrow{y_1}) \vee \cdots \vee \exists \overrightarrow{y_m} B_m(\vec{x}, \overrightarrow{y_m}))$$

such that

- Each A_i an atomic first-order formula
  E.g. $Rxyz$, $y = z$, Ea, $x \leq y$.
- Each B_i a finite conjunction of atomic first-order formulae
  Eg. $Rxyz \wedge y = z$, $Rxyz \wedge Ryzw \wedge Rzwv$.

A Useful Property of Separation Theories

Definition

A **coherent formula** is a first-order formula of the form

$$\forall \vec{x}(A_1(\vec{x}) \wedge \cdots \wedge A_n(\vec{x}) \rightarrow \exists \overrightarrow{y_1} B_1(\vec{x}, \overrightarrow{y_1}) \vee \cdots \vee \exists \overrightarrow{y_m} B_m(\vec{x}, \overrightarrow{y_m}))$$

such that

- Each A_i an atomic first-order formula
  E.g. $Rxyz$, $y = z$, Ea, $x \leq y$.
- Each B_i a finite conjunction of atomic first-order formulae
  Eg. $Rxyz \wedge y = z$, $Rxyz \wedge Ryzw \wedge Rzwv$.

Why important? All* separation properties found in the literature are coherent formulae!

A Useful Property of Separation Theories

Definition

A **coherent formula** is a first-order formula of the form

$$\forall \vec{x}(A_1(\vec{x}) \wedge \cdots \wedge A_n(\vec{x}) \rightarrow \exists \overrightarrow{y_1} B_1(\vec{x}, \overrightarrow{y_1}) \vee \cdots \vee \exists \overrightarrow{y_m} B_m(\vec{x}, \overrightarrow{y_m}))$$

such that

- Each A_i an atomic first-order formula
  E.g. $Rxyz$, $y = z$, Ea, $x \leq y$.
- Each B_i a finite conjunction of atomic first-order formulae
  Eg. $Rxyz \wedge y = z$, $Rxyz \wedge Ryzw \wedge Rzwv$.

Why important? All* separation properties found in the literature are coherent formulae!

Simpson, Braüner, Negri: coherent formulae generate natural deduction/sequent rules...

Coherent Formulae are Tableaux Rules!

Non-branching — $\forall x, y, y'(x \leq y \wedge x \leq y' \rightarrow y \leq y' \vee y' \leq y)$

Coherent Formulae are Tableaux Rules!

Non-branching — $\forall x, y, y'(x \leq y \wedge x \leq y' \rightarrow y \leq y' \vee y' \leq y)$

$$\frac{x \sim y, x \sim y' \in C}{\langle \emptyset, \{y \sim y'\} \rangle \mid \langle \emptyset, \{y' \sim y\} \rangle}$$

Coherent Formulae are Tableaux Rules!

Non-branching — $\forall x, y, y'(x \leq y \wedge x \leq y' \rightarrow y \leq y' \vee y' \leq y)$

$$\frac{x \sim y, x \sim y' \in C}{\langle \emptyset, \{y \sim y'\} \rangle \mid \langle \emptyset, \{y' \sim y\} \rangle}$$

Total — $\forall x, y(\exists z(z \in x \circ y))$

Coherent Formulae are Tableaux Rules!

Non-branching — $\forall x, y, y'(x \leq y \wedge x \leq y' \rightarrow y \leq y' \vee y' \leq y)$

$$\frac{x \sim y, x \sim y' \in C}{\langle \emptyset, \{y \sim y'\} \rangle \mid \langle \emptyset, \{y' \sim y\} \rangle}$$

Total — $\forall x, y(\exists z(z \in x \circ y))$

$$\frac{x, y \text{ occur on branch}}{\langle \emptyset, \{R_* xyc_i\} \rangle}$$

Coherent Formulae are Tableaux Rules!

Non-branching — $\forall x, y, y'(x \leq y \wedge x \leq y' \rightarrow y \leq y' \vee y' \leq y)$

$$\frac{x \sim y, x \sim y' \in C}{\langle \emptyset, \{y \sim y'\} \rangle \mid \langle \emptyset, \{y' \sim y\} \rangle}$$

Total — $\forall x, y(\exists z(z \in x \circ y))$

$$\frac{x, y \text{ occur on branch}}{\langle \emptyset, \{R_* xyc_i\} \rangle}$$

This translation can be done systematically.

A Tableau Proof of $(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \rightarrow \mathrm{I}$ in Total BBI Models

$(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \rightarrow \mathrm{I}$ is valid in Total BBI models

A Tableau Proof of $(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \rightarrow \mathrm{I}$ in Total BBI Models

$(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \rightarrow \mathrm{I}$ is valid in Total BBI models $(\forall x, y(\exists z(z \in x \circ y)))$.

A Tableau Proof of $(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \to \mathrm{I}$ in Total BBI Models

$(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \to \mathrm{I}$ is valid in Total BBI models $(\forall x, y(\exists z(z \in x \circ y)))$.

(1)	$\langle\{\mathbb{F}(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \to \mathrm{I} : c_0\}, \emptyset\rangle$		Premiss
(2)	$\langle\{\mathbb{T}\neg \mathrm{I} \mathrel{-\!\!*} \bot : c_0, \mathbb{F}\mathrm{I} : c_0\}, \emptyset\rangle$		$\langle\mathbb{F} \to\rangle$, from (1)
(3)	$\langle\emptyset, \{R_* c_0 c_0 c_1\}\rangle$		Total, from (1)
(4)	$\langle\{\mathbb{F}\neg \mathrm{I} : c_0\}, \emptyset\rangle$	$\langle\{\mathbb{T}\bot : c_1\}, \emptyset\rangle$	$\langle\mathbb{T}\mathrel{-\!\!*}\rangle$, from (2), (3)
(5)	$\langle\{\mathbb{T}\mathrm{I} : c_0\}, \emptyset\rangle$	$\otimes$	$\langle\mathbb{F}\neg\rangle$, from (4)
(6)	$\langle\emptyset, \{c_0 \sim c_0\}\rangle$		$\langle$Ref$\rangle$, from (5)
	$\otimes$		

A Tableau Proof of $(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \to \mathrm{I}$ in Total BBI Models

$(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \to \mathrm{I}$ is valid in Total BBI models ($\forall x, y(\exists z(z \in x \circ y))$).

(1)	$\langle\{\mathbb{F}(\neg \mathrm{I} \mathrel{-\!\!*} \bot) \to \mathrm{I} : c_0\}, \emptyset\rangle$	Premiss
(2)	$\langle\{\mathbb{T}\neg \mathrm{I} \mathrel{-\!\!*} \bot : c_0, \mathbb{F}\mathrm{I} : c_0\}, \emptyset\rangle$	$\langle\mathbb{F} \to\rangle$, from (1)
(3)	$\langle\emptyset, \{R_* c_0 c_0 c_1\}\rangle$	Total, from (1)
(4)	$\langle\{\mathbb{F}\neg \mathrm{I} : c_0\}, \emptyset\rangle$ | $\langle\{\mathbb{T}\bot : c_1\}, \emptyset\rangle$	$\langle\mathbb{T}\mathrel{-\!\!*}\rangle$, from (2), (3)
(5)	$\langle\{\mathbb{T}\mathrm{I} : c_0\}, \emptyset\rangle$ | $\otimes$	$\langle\mathbb{F}\neg\rangle$, from (4)
(6)	$\langle\emptyset, \{c_0 \sim c_0\}\rangle$	$\langle$Ref$\rangle$, from (5)
	$\otimes$	

Left branch closed: $\mathbb{F}\mathrm{I} : c_0$ at (2), $\mathbb{T}\mathrm{I} : c_0$ at (5) and $c_0 \sim c_0$ at (6).
Right branch closed: $\mathbb{T}\bot : c_1$ at (4).

Metatheory

Tableaux Rules are Coherent Formulae!

Consider a first-order language with predicates $\mathbb{T}\varphi$, $\mathbb{F}\varphi$ (for every bunched logic formula φ), $\circ$, $\leq$ and E.

Tableaux Rules are Coherent Formulae!

Consider a first-order language with predicates $\mathbb{T}\varphi$, $\mathbb{F}\varphi$ (for every bunched logic formula φ), $\circ$, $\leq$ and E.

$$\frac{\mathbb{T}\varphi * \psi : x \in \mathcal{F}}{\langle \{\mathbb{T}\varphi : c_i, \mathbb{T}\psi : c_j\}, \{R_* c_i c_j x\} \rangle}$$

Tableaux Rules are Coherent Formulae!

Consider a first-order language with predicates $\mathbb{T}\varphi$, $\mathbb{F}\varphi$ (for every bunched logic formula φ), $\circ$, $\leq$ and E.

$$\frac{\mathbb{T}\varphi * \psi : x \in \mathcal{F}}{\langle\{\mathbb{T}\varphi : c_i, \mathbb{T}\psi : c_j\}, \{R_* c_i c_j x\}\rangle}$$

$$\Downarrow$$

$$\forall x(\mathbb{T}(\varphi * \psi)(x) \rightarrow \exists y, z(\mathbb{T}(\varphi)(y) \wedge \mathbb{T}(\psi)(z) \wedge x \in y \circ z))$$

Tableaux Rules are Coherent Formulae!

Consider a first-order language with predicates $\mathbb{T}\varphi$, $\mathbb{F}\varphi$ (for every bunched logic formula φ), $\circ$, $\leq$ and E.

$$\frac{\mathbb{T}\varphi * \psi : x \in \mathcal{F}}{\langle\{\mathbb{T}\varphi : c_i, \mathbb{T}\psi : c_j\}, \{R_* c_i c_j x\}\rangle}$$

$$\Downarrow$$

$$\forall x(\mathbb{T}(\varphi * \psi)(x) \rightarrow \exists y, z(\mathbb{T}(\varphi)(y) \wedge \mathbb{T}(\psi)(z) \wedge x \in y \circ z))$$

This translation can be done systematically.

Tableaux Systems as Coherent Theories

Given bunched logic formula φ, obtain coherent theory $\Phi_{\varphi}^{(B)BI+\Sigma}$ from

Tableaux Systems as Coherent Theories

Given bunched logic formula φ, obtain coherent theory $\Phi_{\varphi}^{(B)BI+\Sigma}$ from

- translations of $\mathbb{T}$,$\mathbb{F}$ rule instances for φ subformulae;

Tableaux Systems as Coherent Theories

Given bunched logic formula φ, obtain coherent theory $\Phi_{\varphi}^{(B)BI+\Sigma}$ from

- translations of $\mathbb{T}$,$\mathbb{F}$ rule instances for φ subformulae;
- translations of all constraint rules ((B)BI model rules + Σ rules);

Tableaux Systems as Coherent Theories

Given bunched logic formula φ, obtain coherent theory $\Phi_{\varphi}^{(B)BI+\Sigma}$ from

- translations of $\mathbb{T}$,$\mathbb{F}$ rule instances for φ subformulae;
- translations of all constraint rules ((B)BI model rules + Σ rules);
- translations of all closure conditions.

Tableaux Systems as Coherent Theories

Given bunched logic formula φ, obtain coherent theory $\Phi_{\varphi}^{(B)BI+\Sigma}$ from

- translations of $\mathbb{T}$,$\mathbb{F}$ rule instances for φ subformulae;
- translations of all constraint rules ((B)BI model rules + Σ rules);
- translations of all closure conditions.

Theorem

There exists a (B)BI + Σ-countermodel to φ iff there exists a first-order model of $\Phi_{\varphi}^{(B)BI+\Sigma} \cup \{\exists x.\mathbb{F}\varphi(x)\}$.

(First-order model: carrier set with interpretation of all predicates in language making all φ true)

Parametric Soundness and Completeness

Bezem/Coquand[16]: proof system deriving judgements $X[\vec{a}] \Vdash^{\Phi} D$ for Φ a coherent theory, $\forall \vec{x}(\bigwedge X[\vec{x}] \rightarrow D)$ a coherent formula.

[16] M. Bezem and T. Coquand. Automating Coherent Logic. *LPAR '05*.

Parametric Soundness and Completeness

Bezem/Coquand[16]: proof system deriving judgements $X[\vec{a}] \Vdash^{\Phi} D$ for Φ a coherent theory, $\forall \vec{x}(\bigwedge X[\vec{x}] \rightarrow D)$ a coherent formula.

Theorem

φ is (B)BI + Σ tableaux provable.

φ valid in (B)BI + Σ models.

[16] M. Bezem and T. Coquand. Automating Coherent Logic. *LPAR '05*.

Parametric Soundness and Completeness

Bezem/Coquand[16]: proof system deriving judgements $X[\vec{a}] \Vdash^{\Phi} D$ for Φ a coherent theory, $\forall \vec{x}(\bigwedge X[\vec{x}] \rightarrow D)$ a coherent formula.

Theorem

φ is (B)BI + Σ tableaux provable.

$$\Updownarrow$$

$\{\mathbb{F}(\varphi)(a)\} \Vdash^{\Phi_{\varphi}^{(B)BI+\Sigma}} \bot$ *derivable.*

φ valid in (B)BI + Σ models.

[16] M. Bezem and T. Coquand. Automating Coherent Logic. *LPAR '05*.

Parametric Soundness and Completeness

Bezem/Coquand[16]: proof system deriving judgements $X[\vec{a}] \Vdash^{\Phi} D$ for Φ a coherent theory, $\forall \vec{x}(\bigwedge X[\vec{x}] \rightarrow D)$ a coherent formula.

Theorem

$$\varphi \textit{ is (B)BI} + \Sigma \textit{ tableaux provable.}$$

$$\Updownarrow$$

$$\{\mathbb{F}(\varphi)(a)\} \Vdash^{\Phi_{\varphi}^{(B)BI+\Sigma}} \bot \textit{ derivable.}$$

$$\Updownarrow$$

$$\textit{Exists no first-order model of } \Phi_{\varphi}^{(B)BI+\Sigma} \cup \{\exists x.\mathbb{F}\varphi(x)\}$$

$$\varphi \textit{ valid in (B)BI} + \Sigma \textit{ models.}$$

[16] M. Bezem and T. Coquand. Automating Coherent Logic. *LPAR '05*.

Parametric Soundness and Completeness

Bezem/Coquand[16]: proof system deriving judgements $X[\vec{a}] \Vdash^{\Phi} D$ for Φ a coherent theory, $\forall \vec{x}(\bigwedge X[\vec{x}] \rightarrow D)$ a coherent formula.

Theorem

$$\varphi \textit{ is (B)BI} + \Sigma \textit{ tableaux provable.}$$

$$\Updownarrow$$

$$\{\mathbb{F}(\varphi)(a)\} \Vdash^{\Phi_{\varphi}^{(B)BI+\Sigma}} \bot \textit{ derivable.}$$

$$\Updownarrow$$

$$\textit{Exists no first-order model of } \Phi_{\varphi}^{(B)BI+\Sigma} \cup \{\exists x.\mathbb{F}\varphi(x)\}$$

$$\Updownarrow$$

$$\varphi \textit{ valid in (B)BI} + \Sigma \textit{ models.}$$

[16] M. Bezem and T. Coquand. Automating Coherent Logic. *LPAR '05*.

Conclusions

Conclusions

- Separation Logic = bunched logic verification formalism.

Conclusions

- Separation Logic = bunched logic verification formalism.
- Expressivity gap between bunched logic and the different models of separation logic — proof theory tricky.

Conclusions

- Separation Logic = bunched logic verification formalism.
- Expressivity gap between bunched logic and the different models of separation logic — proof theory tricky.
- Our solution: proof systems that can be systematically extended in a modular fashion.

Conclusions

- Separation Logic = bunched logic verification formalism.
- Expressivity gap between bunched logic and the different models of separation logic — proof theory tricky.
- Our solution: proof systems that can be systematically extended in a modular fashion.
- Sound/complete for (B)BI + any separation theory.

Conclusions

- Separation Logic = bunched logic verification formalism.
- Expressivity gap between bunched logic and the different models of separation logic — proof theory tricky.
- Our solution: proof systems that can be systematically extended in a modular fashion.
- Sound/complete for (B)BI + any separation theory.
- Tableaux systems ⇔ coherent theories.

Conclusions

- Separation Logic = bunched logic verification formalism.
- Expressivity gap between bunched logic and the different models of separation logic — proof theory tricky.
- Our solution: proof systems that can be systematically extended in a modular fashion.
- Sound/complete for (B)BI + any separation theory.
- Tableaux systems ⇔ coherent theories.
- Applicable to: existing bunched/separation logics, many more modal and substructural logics.

Further Work

- Implementation of the tableaux systems: directly or via coherent logic provers[17].

[17] A. Polonksy. Proofs, Types and Lambda Calculus. Univ. Bergen, 2012.

Further Work

- Implementation of the tableaux systems: directly or via coherent logic provers[17].
- Proof-search strategies: blocking for BI, constraint limiting.

[17] A. Polonksy. Proofs, Types and Lambda Calculus. Univ. Bergen, 2012.

Further Work

- Implementation of the tableaux systems: directly or via coherent logic provers[17].
- Proof-search strategies: blocking for BI, constraint limiting.
- Parametric Separation Logic implementations.

[17] A. Polonksy. Proofs, Types and Lambda Calculus. Univ. Bergen, 2012.

Further Work

- Implementation of the tableaux systems: directly or via coherent logic provers[17].
- Proof-search strategies: blocking for BI, constraint limiting.
- Parametric Separation Logic implementations.
- Logical frameworks: generic logic with generic countermodel generation based on this approach.

[17] A. Polonksy. Proofs, Types and Lambda Calculus. Univ. Bergen, 2012.