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\textsuperscript{1}

\textsuperscript{1}S. Docherty and D. Pym. Modular Tableaux Calculi for Separation Theories. \textit{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\(^2\):
  \[ \varphi * \varphi \not\equiv \varphi. \]

---


\(^3\)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\(^2\):

  \[ \varphi \ast \varphi \not\equiv \varphi. \]

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

---


\(^3\)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\(^2\):
  \[ \varphi \cdot \varphi \not\equiv \varphi. \]

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

---


\(^3\)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\(^2\): 
  \[ \varphi \ast \varphi \not\equiv \varphi. \]

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

---


\(^3\)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 $\text{Prop}$.

$$\varphi ::= p \mid \top \mid \bot \mid I \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \varphi \rightarrow \varphi \mid \varphi \ast \varphi \mid \varphi \astast \varphi$$

---


The Logic of Bunched Implications

Let $p$ range over set of propositional atoms $\text{Prop}$.

$$\varphi ::= p \mid \top \mid \bot \mid I \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \varphi \rightarrow \varphi \mid \varphi^* \mid \varphi^*$$

- **BI**: standard connectives = intuitionistic logic; decidable

---


The Logic of Bunched Implications

Let \( p \) range over set of propositional atoms Prop.

\[
\varphi ::= p \mid \top \mid \bot \mid I \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \varphi \rightarrow \varphi \mid \varphi \ast \varphi \mid \varphi \ast \ast \varphi
\]

- **BI:** standard connectives = intuitionistic logic; decidable\(^4\)
- **Boolean BI (BBI):** standard connectives = classical logic; undecidable\(^5\)

---


The Logic of Bunched Implications

Let $p$ range over set of propositional atoms $Prop$.

$$
\varphi ::= p \mid \top \mid \bot \mid I \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \varphi \to \varphi \mid \varphi * \varphi \mid \varphi \dashv \varphi
$$

- BI: standard connectives $=$ intuitionistic logic; decidable$^4$
- Boolean BI (BBI): standard connectives $=$ classical logic; undecidable$^5$
- $*$ and $\dashv$ adjoint: $(\varphi * \psi) \to \chi \equiv \varphi \to (\psi \dashv \chi)$;
- $*$ associative: $(\varphi * \psi) * \chi \equiv \varphi * (\psi * \chi)$;
- $*$ commutative: $\varphi * \psi \equiv \psi * \varphi$;
- $I$ is unit for $*$: $\varphi * I \equiv \varphi$;


Resource Semantics

Definition (BI Model)

- Set of resources $X$;
Resource Semantics

Definition (**BI Model**)

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

**Definition (BI Model)**

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

Definition (*BI Model*)

- Set of resources $X$;
- Composition $\circ : X^2 \rightarrow \mathcal{P}(X)$, 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 \( \mathcal{V} : \text{Prop} \to \mathcal{P}(X) \).
Resource Semantics

**Definition (BI Model)**

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

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 \( \mathcal{V} : \text{Prop} \to \mathcal{P}(X) \).

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

- \( x \vDash I \) iff \( x \in E \).
- \( x \vDash \varphi \ast \psi \) 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 $V : \text{Prop} \to \mathcal{P}(X)$.

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

- $x \not\in I$ iff $x \in E$.
- $x \in \varphi \ast \psi$ 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\(^6\).

---

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, \(\preceq\) is heap extension.

---

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, \(\preceq\) is heap extension.
- Calculus of Hoare triples \(\{\varphi\}C\{\psi\}\): if memory in state \(\varphi\) and \(C\) executes, memory will be in state \(\psi\) afterwards.

---

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 \(\varphi\) and \(C\) executes, memory will be in state \(\psi\) afterwards.
- Frame rule \(\Rightarrow\) scalability:

\[
\frac{\{\varphi\}C\{\psi\}}{\{\varphi \ast \chi\}C\{\psi \ast \chi\}},
\]

where \(C\) doesn’t modify memory described by \(\chi\).

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 \(\varphi\) and \(C\) executes, memory will be in state \(\psi\) afterwards.
- Frame rule \(\Rightarrow\) scalability:

  \[
  \frac{\{\varphi\} C \{\psi\}}{\{\varphi * \chi\} C \{\psi * \chi\}},
  \]

  where \(C\) doesn’t modify memory described by \(\chi\).
- Different memory models for different tasks \(\Rightarrow\) zoo of separation logics.

A Taxonomy of Separation Logics: Separation Theories

Abstraction of separation logics as Separation Algebras\(^7\)\(^8\)\(^9\):
(B)BI models in which ≤, ○, 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.*

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**
  \[ \forall x, y, z, z'(z \in x \circ y \land z' \in x \circ y \rightarrow z = z') \]

- **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** \(\forall x, y, z, z'(z \in x \circ y \land z' \in x \circ y \rightarrow z = z')\)
- **Total** \(\forall x, y(\exists z(z \in x \circ y))\)
- **Single Unit** \(\forall x, x'(x \in E \land x' \in E \rightarrow x = x')\)
- **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** \(\forall x, y, z, z'(z \in x \circ y \land z' \in x \circ y \rightarrow z = z')\)
- **Total** \(\forall x, y(\exists z(z \in x \circ y))\)
- **Single Unit** \(\forall x, x'(x \in E \land x' \in E \rightarrow x = x')\)
- **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** \(\forall x, y, y'(x \leq y \land x \leq y' \rightarrow y \leq y' \lor y' \leq y)\)
- **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.*
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 $\varphi$ such that

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

\footnote{J. Brotherston and J. Villard. Parametric Completeness for Separation Theories. \textit{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 $\varphi$ 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*\(^{10}\).

Some Separation Theories Are Undefinable in (B)BI

Given logic \( L \), a model property \( P \) is \( L \)-definable if there exists an \( L \) – formula \( \varphi \) 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*\(^{10}\).

---

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\textsuperscript{11}.

Proof Sketch.

\((\neg I \to \bot) \to I\) is valid for all Total models, but there exists a Partial model for which it is not. \(\square\)

---

\textsuperscript{11}D. Larchey-Wendling and D. Galmiche. The Undecidability of Boolean BI Through Phase Semantics. \textit{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.

\((\neg I \multimap \bot) \rightarrow I\) is valid for all Total models, but there exists a Partial model for which it is not.

\[\square\]

Note: \((\neg I \multimap \bot) \rightarrow I\) 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.

---


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.

---


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\textsuperscript{12} and/or restricted in application\textsuperscript{13}.

\textsuperscript{12} J. Brotherston and J. Villard. Parametric Completeness for Separation Theories. \textit{POPL ’14}.

\textsuperscript{13} Z. Hoú, R. Clouston, A. Tiu and R. Goré. Proof Search for Propositional Abstract Separation Logic with Labelled Sequents. \textit{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\(^\text{12}\) and/or restricted in application\(^\text{13}\).
- **We give a modular tableaux system capturing validity in all (B)BI + $\Sigma$ models, for any separation theory $\Sigma$.**

---


\(^{13}\) 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: $S \in \{T, F\}$ (true/false);
Basic Units: Labelled Formulae and Constraints

Two basic syntactic entities:

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

Two basic syntactic entities:

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

2. **Constraints** on labels encoding a (partial) model
Basic Units: Labelled Formulae and Constraints

Two basic syntactic entities:

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

2. **Constraints** on labels encoding a (partial) model
   - $x \sim y$: resource $x \leq$ resource $y$;
   - $R_* yzx$: resource $x \in$ resource $y$ ◦ resource $z$;
Basic Units: Labelled Formulae and Constraints

Two basic syntactic entities:

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

2. **Constraints** on labels encoding a (partial) model
   - $x \sim y$: resource $x \leq$ resource $y$;
   - $R_\ast yzx$: 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\(^\text{14} \ 15\)


How Our Proof Systems Work: Rules, Tableaux, Proofs

- **Tableau** = tree with nodes given by finite sets of labelled formulae and constraints\(^{14}\) \(^{15}\)
  - Root = \(\{F\varphi : c_0\}\) where \(\varphi\) is formula trying to prove.

---


How Our Proof Systems Work: Rules, Tableaux, Proofs

- **Tableau** = tree with nodes given by finite sets of labelled formulae and constraints\(^\text{14}\)\(^\text{15}\)

- Root = \(\{F\varphi : c_0\}\) where \(\varphi\) is formula trying to prove.

- Constructed by rules

\[
\frac{\$\psi : x \in F \text{ and } \gamma_1, \ldots, \gamma_n \in C}{\langle F_1, C_1 \rangle | \ldots | \langle F_k, C_k \rangle}
\]

---


How Our Proof Systems Work: Rules, Tableaux, Proofs

- **Tableau** = tree with nodes given by finite sets of labelled formulae and constraints\(^{14}\) \(^{15}\)
- **Root** = \(\{F\varphi : c_0\}\) where \(\varphi\) is formula trying to prove.
- **Constructed by rules**

\[
\begin{align*}
\text{If } S\psi : x &\in F \text{ and } \gamma_1, \ldots, \gamma_n \in C \\
\text{branch extends into } k \text{ branches through addition of formulae/constraints } \langle F_i, C_i \rangle
\end{align*}
\]

"If \(S\psi : x\) and constraints \(\gamma_i\) on branch, branch extends into \(k\) branches through addition of formulae/constraints \(\langle F_i, C_i \rangle\)."

---


How Our Proof Systems Work: Rules, Tableaux, Proofs

- **Tableau** = tree with nodes given by finite sets of labelled formulae and constraints\(^{14}\) \(^{15}\)
- **Root** = \(\{F\varphi : c_0\}\) where \(\varphi\) is formula trying to prove.
- Constructed by rules

\[
\frac{\exists \psi : x \in F \text{ and } \gamma_1, \ldots, \gamma_n \in C}{\langle F_1, C_1 \rangle \mid \ldots \mid \langle F_k, C_k \rangle}
\]

“If \(\exists \psi : x\) and constraints \(\gamma_i\) on branch, branch extends into \(k\) branches through addition of formulae/constraints \(\langle F_i, C_i \rangle\).”

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


Basic Tableaux Rules for Bunched Logic

- Each connective has $T$ and $F$ rules for decomposing formulae.
Basic Tableaux Rules for Bunched Logic

- Each connective has $T$ and $F$ rules for decomposing formulae.

Examples for BBI:

\[
\begin{align*}
\langle T^* \rangle & \quad T\varphi \ast \psi : x \in F \\
\{T\varphi : c_i, T\psi : c_j\}, \{R^* c_i c_j x\} & \\
\langle F^* \rangle & \quad F\varphi \ast \psi : x \in F \text{ and } R^* yzx \in C \\
\{F\varphi : y\}, \emptyset & \quad | \quad \{F\psi : z\}, \emptyset
\end{align*}
\]
Basic Tableaux Rules for Bunched Logic

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

\[
\begin{align*}
\langle \mathbb{T}^* \rangle & \quad \frac{\mathbb{T} \varphi \ast \psi : x \in \mathcal{F}}{\langle \{ \mathbb{T} \varphi : c_i, \mathbb{T} \psi : c_j \}, \{ R^* c_i c_j x \} \rangle} \\
\langle \mathbb{F}^* \rangle & \quad \frac{\mathbb{F} \varphi \ast \psi : x \in \mathcal{F} \text{ and } R^* yzx \in \mathcal{C}}{\langle \{ \mathbb{F} \varphi : y \}, \emptyset \rangle \mid \langle \{ \mathbb{F} \psi : z \}, \emptyset \rangle}
\end{align*}
\]

- Rules acting on constraints.
Basic Tableaux Rules for Bunched Logic

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

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

- Rules acting on constraints.
  Examples for BBI:

  $$\langle \text{Comm} \rangle \quad \frac{R^* xyz \in C}{\langle \emptyset, \{R^* yzx\} \rangle} \quad \langle \text{Sym} \rangle \quad \frac{x \sim y \in C}{\langle \emptyset, \{y \sim x\} \rangle}$$
A Useful Property of Separation Theories

Definition

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

$$\forall \bar{x}(A_1(\bar{x}) \land \cdots \land A_n(\bar{x}) \rightarrow \exists \bar{y} B_1(\bar{x}, \bar{y}) \lor \cdots \lor \exists \bar{y} B_m(\bar{x}, \bar{y}))$$

such that

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

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

Simpson, Braüner, Negri: coherent formulae generate natural deduction/sequent rules...
A Useful Property of Separation Theories

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

\[ \forall \bar{x}(A_1(\bar{x}) \land \cdots \land A_n(\bar{x}) \rightarrow \exists \bar{y} B_1(\bar{x}, \bar{y}) \lor \cdots \lor \exists \bar{y} B_m(\bar{x}, \bar{y})) \]

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

\[ \forall \vec{x}(A_1(\vec{x}) \land \cdots \land A_n(\vec{x}) \rightarrow \exists \vec{y}_1 B_1(\vec{x}, \vec{y}_1) \lor \cdots \lor \exists \vec{y}_m B_m(\vec{x}, \vec{y}_m)) \]

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

\[ \forall \bar{x}(A_1(\bar{x}) \land \cdots \land A_n(\bar{x}) \rightarrow \exists \bar{y}_1 B_1(\bar{x}, \bar{y}_1) \lor \cdots \lor \exists \bar{y}_m B_m(\bar{x}, \bar{y}_m)) \]

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

$$\forall \bar{x}(A_1(\bar{x}) \land \cdots \land A_n(\bar{x}) \rightarrow \exists \bar{y} \land B_1(\bar{x}, \bar{y}) \lor \cdots \lor \exists \bar{y} \land B_m(\bar{x}, \bar{y}))$$

such that

- Each $A_i$ an atomic first-order formula
  E.g. $R_{xyz}, y = z, E_a, x \leq y$.
- Each $B_i$ a finite conjunction of atomic first-order formulae
  Eg. $R_{xyz} \land y = z, R_{xyz} \land R_{yzw} \land R_{zwv}$. 

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

Simpson, Braüner, Negri: coherent formulae generate natural deduction/sequent rules...
A Useful Property of Separation Theories

**Definition**

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

$$\forall \bar{x}(A_1(\bar{x}) \land \cdots \land A_n(\bar{x}) \rightarrow \exists \bar{y}_1 B_1(\bar{x}, \bar{y}_1) \lor \cdots \lor \exists \bar{y}_m B_m(\bar{x}, \bar{y}_m))$$

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
  
  E.g. $Rxyz \land y = z, Rxyz \land Ryzw \land Rzwv$.

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

\[ \forall \bar{x}(A_1(\bar{x}) \land \cdots \land A_n(\bar{x}) \rightarrow \exists \bar{y} B_1(\bar{x}, \bar{y}_1) \lor \cdots \lor \exists \bar{y}_m B_m(\bar{x}, \bar{y}_m)) \]

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
  
  E.g. \( Rxyz \land y = z, Rxyz \land Ryzw \land Rzwv \).

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 — \( \forall x, y, y' (x \leq y \land x \leq y' \rightarrow y \leq y' \lor y' \leq y) \)
Coherent Formulae are Tableaux Rules!

Non-branching — $\forall x, y, y'(x \leq y \land x \leq y' \rightarrow y \leq y' \lor y' \leq y)$

\[
\frac{x \sim y, x \sim y' \in C}{\langle \emptyset, \{y \sim y'\} \rangle | \langle \emptyset, \{y' \sim y\} \rangle}
\]
Coherent Formulae are Tableaux Rules!

Non-branching — \( \forall x, y, y'(x \leq y \land x \leq y' \rightarrow y \leq y' \lor y' \leq y) \)

\[
\frac{x \sim y, x \sim y' \in C}{\langle \emptyset, \{y \sim y'\} \rangle | \langle \emptyset, \{y' \sim y\} \rangle}
\]

Total — \( \forall x, y(\exists z(z \in x \circ y)) \)
Coherent Formulae are Tableaux Rules!

Non-branching — \( \forall x, y, y'(x \leq y \land x \leq y' \rightarrow y \leq y' \lor y' \leq y) \)

\[
\begin{align*}
\frac{x \sim y, x \sim y' \in C}{\langle \emptyset, \{y \sim y'\} \rangle \mid \langle \emptyset, \{y' \sim y\} \rangle}
\end{align*}
\]

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 — \( \forall x, y, y'(x \leq y \land x \leq y' \rightarrow y \leq y' \lor y' \leq y) \)

\[
\frac{x \sim y, x \sim y' \in C}{\langle \emptyset, \{y \sim y'\} \rangle | \langle \emptyset, \{y' \sim y\} \rangle}
\]

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 \( (\neg I \star \bot) \rightarrow I \) in Total BBI Models

\((\neg I \star \bot) \rightarrow I\) is valid in Total BBI models
A Tableau Proof of \((\neg I \rightarrow^* \bot) \rightarrow I\) in Total BBI Models

\((\neg I \rightarrow^* \bot) \rightarrow I\) is valid in Total BBI models \((\forall x, y(\exists z(z \in x \circ y)))\).
A Tableau Proof of \((\neg I \ast \bot) \rightarrow I\) in Total BBI Models

\((\neg I \ast \bot) \rightarrow I\) is valid in Total BBI models \((\forall x, y(\exists z(z \in x \circ y)))\).

\[
\begin{align*}
(1) & \quad \langle \{F(\neg I \ast \bot) \rightarrow I : c_0\}, \emptyset \rangle & \text{Premiss} \\
(2) & \quad \langle \{T \neg I \ast \bot : c_0, FI : c_0\}, \emptyset \rangle & \langle F \rightarrow \rangle, \text{from (1)} \\
(3) & \quad \langle \emptyset, \{R_\ast c_0 c_0 c_1\} \rangle & \langle \text{Total}, \text{from (1)} \rangle \\
(4) & \quad \langle \{F \neg I : c_0\}, \emptyset \rangle & \langle \{T \bot : c_1\}, \emptyset \rangle & \langle T \ast \rangle, \text{from (2), (3)} \\
(5) & \quad \langle \{TI : c_0\}, \emptyset \rangle & \langle F \neg \rangle, \text{from (4)} \\
(6) & \quad \langle \emptyset, \{c_0 \sim c_0\} \rangle & \langle \text{Ref}, \text{from (5)} \rangle \\
\end{align*}
\]
A Tableau Proof of \((\neg I \Rightarrow \bot) \rightarrow I\) in Total BBI Models

\((\neg I \Rightarrow \bot) \rightarrow I\) is valid in Total BBI models \((\forall x, y(\exists z(z \in x \circ y)))\).

\begin{align*}
(1) & \quad \langle \{F(\neg I \Rightarrow \bot) \rightarrow I : c_0\}, \emptyset \rangle \quad \text{Premiss} \\
(2) & \quad \langle \{T \neg I \Rightarrow \bot : c_0, FI : c_0\}, \emptyset \rangle \quad \langle F \rightarrow \rangle, \text{from (1)} \\
(3) & \quad \langle \emptyset, \{R_*c_0c_0c_1\} \rangle \quad \text{Total, from (1)}
\end{align*}

\begin{align*}
(4) & \quad \langle \{\neg F I : c_0\}, \emptyset \rangle \quad \langle \top \bot : c_1\}, \emptyset \rangle \quad \langle T_* \rangle, \text{from (2), (3)} \\
(5) & \quad \langle \top I : c_0\}, \emptyset \rangle \quad \otimes \quad \langle F \neg \rangle, \text{from (4)} \\
(6) & \quad \langle \emptyset, \{c_0 \sim c_0\} \rangle \quad \otimes \quad \langle \text{Ref} \rangle, \text{from (5)}
\end{align*}

Left branch closed: \(FI : c_0\) at (2), \(TI : c_0\) at (5) and \(c_0 \sim c_0\) at (6).
Right branch closed: \(T \bot : c_1\) at (4).
Metatheory
Tableaux Rules are Coherent Formulae!

Consider a first-order language with predicates $T\varphi, F\varphi$ (for every bunched logic formula $\varphi$), $\circ, \leq$ and $E$. 
**Tableaux Rules are Coherent Formulae!**

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

$$
T\varphi \ast \psi : x \in F
\begin{array}{c}
\langle \{T\varphi : c_i, T\psi : c_j\}, \{R_c c_i c_j x\}\rangle
\end{array}
$$
Consider a first-order language with predicates $T\varphi$, $F\varphi$ (for every bunched logic formula $\varphi$), $\circ$, $\leq$ and $E$.

$$
\frac{T\varphi \ast \psi : x \in F}{\langle\{T\varphi : c_i, T\psi : c_j\}, \{R_\ast c_i c_j x\}\rangle} \\
\downarrow

\forall x (T(\varphi \ast \psi)(x) \rightarrow \exists y, z (T(\varphi)(y) \land T(\psi)(z) \land x \in y \circ z))$$
Tableaux Rules are Coherent Formulae!

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

\[
T\varphi \ast \psi : x \in F
\]

\[
\langle \{ T\varphi : c_i, T\psi : c_j \}, \{ R_* c_i c_j x \} \rangle
\]

\[
\downarrow
\]

\[
\forall x (T(\varphi \ast \psi)(x) \rightarrow \exists y, z (T(\varphi)(y) \land T(\psi)(z) \land x \in y \circ z))
\]

This translation can be done systematically.
Tableaux Systems as Coherent Theories

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

- translations of $\mathbb{T}, \mathbb{F}$ rule instances for $\varphi$ subformulæ;
Tableaux Systems as Coherent Theories

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

- translations of \( T,F \) rule instances for \( \varphi \) subformulae;
- translations of all constraint rules ((B)BI model rules + \( \Sigma \) rules);

Theorem

There exists a \( (B)BI+\Sigma \)-countermodel to \( \varphi \) iff there exists a first-order model of \( \Phi^{(B)BI+\Sigma}_\varphi \).

(First-order model: carrier set with interpretation of all predicates in language making all \( \varphi \) true)
Tableaux Systems as Coherent Theories

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

- translations of $T,F$ rule instances for $\varphi$ subformulae;
- translations of all constraint rules ($(B)BI$ model rules + $\Sigma$ rules);
- translations of all closure conditions.
Tableaux Systems as Coherent Theories

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

- translations of $\top,\bot$ rule instances for $\varphi$ subformulæ;
- translations of all constraint rules ($(B)BI$ model rules + $\Sigma$ rules);
- translations of all closure conditions.

**Theorem**

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

(First-order model: carrier set with interpretation of all predicates in language making all $\varphi$ true)
Parametric Soundness and Completeness

Bezem/Coquand\textsuperscript{16}: proof system deriving judgements $X[\bar{a}] \models \Phi ~ D$ for $\Phi$ a coherent theory, $\forall \bar{x}(\bigwedge X[\bar{x}] \rightarrow D)$ a coherent formula.

\textsuperscript{16}M. Bezem and T. Coquand. Automating Coherent Logic. \textit{LPAR ’05}. 
Parametric Soundness and Completeness

Bezem/Coquand\textsuperscript{16}: proof system deriving judgements $X[\bar{a}] \vdash \Phi D$ for $\Phi$ a coherent theory, $\forall \bar{x} (\land X[\bar{x}] \rightarrow D)$ a coherent formula.

**Theorem**

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

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

\textsuperscript{16}M. Bezem and T. Coquand. Automating Coherent Logic. LPAR ’05.
Parametric Soundness and Completeness

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

Theorem

\[ \varphi \text{ is } (B)BI + \Sigma \text{ tableaux provable.} \]

\[ \updownarrow \]

\[ \{\mathcal{F}(\varphi)(a)\} \vdash \Phi_{\varphi}^{(B)BI+\Sigma} \perp \text{ derivable.} \]

\[ \varphi \text{ valid in } (B)BI + \Sigma \text{ models.} \]

\textsuperscript{16}M. Bezem and T. Coquand. Automating Coherent Logic. \textit{LPAR '05}. 

Parametric Soundness and Completeness

Bezem/Coquand\(^{16}\): proof system deriving judgements \(X[\bar{a}] \vdash \Phi \Downarrow D\) for \(\Phi\) a coherent theory, \(\forall \bar{x} (\wedge X[\bar{x}] \rightarrow D)\) a coherent formula.

**Theorem**

\(\varphi\) is \((B)BI + \Sigma\) tableaux provable.

\[
\uparrow \Downarrow \quad \{ F(\varphi)(a) \} \vdash \Phi^\varphi_{(B)BI+\Sigma} \Downarrow \text{ derivable.}
\]

Exists no first-order model of \(\Phi^\varphi_{(B)BI+\Sigma} \cup \{ \exists x. F\varphi(x) \}\)

\(\varphi\) valid in \((B)BI + \Sigma\) models.

\(^{16}\)M. Bezem and T. Coquand. Automating Coherent Logic. *LPAR ’05.*
Parametric Soundness and Completeness

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

Theorem

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

\[\upharpoonright \downharpoonleft \{\mathcal{F}(\varphi)(a)\} \vdash \Phi^{(B)BI+\Sigma}_\varphi \downarrow \text{ derivable.}\]

\[\upharpoonright \downharpoonleft \text{Exists no first-order model of } \Phi^{(B)BI+\Sigma}_\varphi \cup \{\exists x.\mathcal{F}\varphi(x)\}\]

\[\upharpoonright \downharpoonleft \varphi \text{ valid in } (B)BI + \Sigma \text{ models.}\]

\textsuperscript{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 $\leftrightarrow$ 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\(^\text{17}\).

\(^{17}\text{A. Polonksy. Proofs, Types and Lambda Calculus. Univ. Bergen, 2012.}\)
Further Work

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

Further Work

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

Further Work

- Implementation of the tableaux systems: directly or via coherent logic provers\textsuperscript{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.