Semantic Cut Elimination for
the Logic of Bunched Implications and Structural Extensions
(as formalized in Coq)

Dan Frumin
Grolog, 13 Oct
Semantic cut elimination for the logic of Bunched Implications with structural extensions, formalized in Coq.
Semantic cut elimination for the logic of Bunched Implications with structural extensions, formalized in Coq.

- **BI**: a logic for reasoning about (separation of) resources.
Semantic cut elimination for the logic of Bunched Implications with structural extensions, formalized in Coq.

- **BI**: a logic for reasoning about (separation of) resources.
- **Cut elimination**: a proof of \( \vdash \varphi \) only includes subformulas of \( \varphi \).
Semantic cut elimination for the logic of Bunched Implications with structural extensions, formalized in Coq.

- **BI**: a logic for reasoning about (separation of) resources.
- Cut elimination: a proof of $\vdash \varphi$ only includes subformulas of $\varphi$.
- **Semantic proof**: proof by interpreting syntax in a model.
Semantic cut elimination for the logic of Bunched Implications with **structural extensions**, formalized in Coq.

- **BI**: a logic for reasoning about (separation of) resources.
- Cut elimination: a proof of $\vdash \varphi$ only includes subformulas of $\varphi$.
- Semantic proof: proof by interpreting syntax in a model.
- **Structural extensions**: extensions of the logic with certain axioms/rules.
Semantic cut elimination for the logic of Bunched Implications with structural extensions, **formalized in Coq**.

- **BI**: a logic for reasoning about (separation of) resources.
- Cut elimination: a proof of $\vdash \varphi$ only includes subformulas of $\varphi$.
- Semantic proof: proof by interpreting syntax in a model.
- Structural extensions: extensions of the logic with certain axioms/rules.
- **Formalized in Coq**: axiom-free formalization at

BI freely combines intuitionistic and linear connectives:

\[ \varphi, \psi \in Prop ::= \text{True} \mid \text{False} \mid \varphi \lor \psi \mid \varphi \land \psi \mid \varphi \rightarrow \psi \]
The logic of Bunched Implications

BI freely combines intuitionistic and linear connectives:

\( \varphi, \psi \in \text{Prop} ::= \text{True} \mid \text{False} \mid \varphi \lor \psi \mid \varphi \land \psi \mid \varphi \to \psi \mid \text{Emp} \mid \varphi \ast \psi \mid \varphi \ast \neg \psi \)

Intuitionistic logic
The logic of Bunched Implications

BI freely combines intuitionistic and linear connectives:

\[ \varphi, \psi \in \text{Prop} ::= \text{True} \mid \text{False} \mid \varphi \lor \psi \mid \varphi \land \psi \mid \varphi \rightarrow \psi \mid \text{Emp} \mid \varphi \ast \psi \mid \varphi \rightarrow \ast \psi \]

Linear logic (fragment)

Propositional logic represents ownership of resources

Intuitionistic logic

Linear logic (fragment)

True \land \varphi = \varphi \land (\varphi \rightarrow \psi) \vdash \psi

Emp \ast \varphi = \varphi \ast (\varphi \rightarrow \ast \psi) \vdash \psi

\varphi \land \psi \vdash \varphi \land \varphi

\varphi \ast \psi \nvdash \varphi \land \varphi

\varphi \rightarrow \ast \psi

\varphi \rightarrow \ast \varphi

Both \varphi and \psi hold for owned resources

\varphi and \psi hold for separate/disjoint resources
BI freely combines intuitionistic and linear connectives:

\[ \varphi, \psi \in Prop ::= \text{True} \mid \text{False} \mid \varphi \lor \psi \mid \varphi \land \psi \mid \varphi \rightarrow \psi \]

| Emp \mid \varphi \ast \psi \mid \varphi \rightarrow \psi |

\[
\begin{align*}
\text{True} \land \varphi &= \varphi \\
\text{Emp} \ast \varphi &= \varphi
\end{align*}
\]

\[
\begin{align*}
\varphi \land (\varphi \rightarrow \psi) &\vdash \psi \\
\varphi \ast (\varphi \rightarrow \psi) &\vdash \psi
\end{align*}
\]

\[
\begin{align*}
\varphi \land \psi &\vdash \varphi \\
\varphi &\vdash \varphi \land \varphi \\
\varphi \ast \psi &\vdash \varphi \\
\varphi &\vdash \varphi \ast \varphi
\end{align*}
\]
The logic of Bunched Implications

BI freely combines intuitionistic and linear connectives:

\( \varphi, \psi \in Prop ::= \text{True} \mid \text{False} \mid \varphi \lor \psi \mid \varphi \land \psi \mid \varphi \rightarrow \psi \)

\mid \text{Emp} \mid \varphi \ast \psi \mid \varphi \ast \psi \)

Proposition represent ownership of resources
The logic of Bunched Implications

BI freely combines intuitionistic and linear connectives:

\[ \varphi, \psi \in \text{Prop} ::= \text{True} \mid \text{False} \mid \varphi \lor \psi \mid \varphi \land \psi \mid \varphi \rightarrow \psi \]

| Emp | \varphi \ast \psi | \varphi \nRightarrow \psi |

Proposition represent ownership of resources.

Both \( \varphi \) and \( \psi \) hold for owned resources.

\( \varphi \) and \( \psi \) hold for separate/disjoint resources.
Why BI?

BI has seen a lot of applications in CS, especially as a basis for program logics for programs with arrays/dynamic memory

- $\ell \mapsto v$: the current state has the location $\ell$ in memory, and it stores the value $v$
- $P \ast Q$: the current state can be divided into two disjoint parts, for which $P$ and $Q$ hold respectively
BI has seen a lot of applications in CS, especially as a basis for *program logics* for programs with arrays/dynamic memory

- $\ell \mapsto v$: the current state has the location $\ell$ in memory, and it stores the value $v$
- $P \ast Q$: the current state can be divided into two disjoint parts, for which $P$ and $Q$ hold respectively
- $\ell \mapsto v \ast \ell' \mapsto v'$: the locations $\ell$ and $\ell'$ do not alias each other
- $\ell \mapsto v \land \ell' \mapsto v'$: aliasing is allowed
Why BI?

BI has seen a lot of applications in CS, especially as a basis for *program logics* for programs with arrays/dynamic memory

- $\ell \mapsto v$: the current state has the location $\ell$ in memory, and it stores the value $v$
- $P \ast Q$: the current state can be divided into two disjoint parts, for which $P$ and $Q$ hold respectively
- $\ell \mapsto v \ast \ell' \mapsto v'$: the locations $\ell$ and $\ell'$ do not alias each other
- $\ell \mapsto v \land \ell' \mapsto v'$: aliasing is allowed
- $\ell_1 \mapsto (v_1, \ell_2) \ast \ell_2 \mapsto (v_2, \ell_3) \ast \cdots \ast \ell_n \mapsto (v_n, \ell_o) \ast \ell_o \mapsto \text{NULL}$: a linked list without cycles
Sequent calculus

Sequent: \( \Gamma \vdash \phi \)

\[
\frac{\Gamma; \varphi; \psi \vdash \chi}{\Gamma; \varphi \land \psi \vdash \chi}
\]

\[
\frac{\Gamma_1 \vdash \varphi \quad \Gamma_2 \vdash \psi}{\Gamma_1; \Gamma_2 \vdash \varphi \land \psi}
\]
Sequent calculus

Left and right rules

\[
\frac{\Gamma; \varphi; \psi \vdash \chi}{\Gamma; \varphi \wedge \psi \vdash \chi}
\]

\[
\frac{\Gamma; \chi \vdash \chi}{\Gamma \vdash \chi}
\]

\[
\frac{\Gamma_1 \vdash \varphi}{\Gamma_1; \Gamma_2 \vdash \varphi \wedge \psi}
\]

\[
\frac{\Gamma_2 \vdash \psi}{\Gamma_1; \Gamma_2 \vdash \varphi \wedge \psi}
\]

\[
\frac{\Gamma \vdash \chi}{\Gamma; \Gamma' \vdash \chi}
\]
Sequent calculus

Left and right rules

Structural rules

Γ; Φ; Ψ ⊬ Ψ
Γ; Φ ⊬ Ψ

Γ; Φ ⊬ Ψ

Γ; Φ; Ψ ⊬ χ
Γ; Φ ⊬ Ψ

Γ; Φ ⊬ Ψ

Γ; Φ; Ψ ⊬ χ
Γ; Φ ⊬ Ψ

Γ; Φ ⊬ Ψ

Γ; Φ; Ψ ⊬ χ
Γ; Φ ⊬ Ψ

Γ; Φ ⊬ Ψ

Γ; Φ; Ψ ⊬ χ
Γ; Φ ⊬ Ψ

Γ; Φ ⊬ Ψ

Γ; Φ; Ψ ⊬ χ
Γ; Φ ⊬ Ψ
Sequent calculus

\[ \frac{\Gamma; (\varphi \; , \; \psi) \vdash \chi}{\Gamma; \varphi \ast \psi \vdash \chi} \]

\[ \frac{\Gamma; \varphi \; ; \; \psi \vdash \chi}{\Gamma; \varphi \land \psi \vdash \chi} \]

\[ \frac{\Gamma \; ; \; \Gamma \vdash \chi}{\Gamma \vdash \chi} \]

\[ \frac{\Gamma_1 \vdash \varphi \quad \Gamma_2 \vdash \psi}{\Gamma_1 \; , \; \Gamma_2 \vdash \varphi \ast \psi} \]

\[ \frac{\Gamma_1 \vdash \varphi \quad \Gamma_2 \vdash \psi}{\Gamma_1 \; ; \; \Gamma_2 \vdash \varphi \land \psi} \]

\[ \frac{\Gamma \vdash \chi}{\Gamma \; ; \; \Gamma' \vdash \chi} \]
Sequent calculus

\[\Delta(\varphi, \psi) \vdash \chi\]
\[\Delta(\varphi \ast \psi) \vdash \chi\]

\[\Delta(\varphi ; \psi) \vdash \chi\]
\[\Delta(\varphi \land \psi) \vdash \chi\]

\[\Delta(\Gamma ; \Gamma) \vdash \chi\]
\[\Delta(\Gamma) \vdash \chi\]

\[\Gamma \vdash \varphi \quad \Gamma \vdash \psi\]
\[\Gamma_1, \Gamma_2 \vdash \varphi \ast \psi\]

\[\Gamma_1 \vdash \varphi \quad \Gamma_2 \vdash \psi\]
\[\Gamma_1 ; \Gamma_2 \vdash \varphi \land \psi\]

\[\Delta(\Gamma) \vdash \chi\]
\[\Delta(\Gamma ; \Gamma') \vdash \chi\]

\[\Gamma ::= \varphi \mid \Gamma ; \Gamma \mid \Gamma, \Gamma \mid \ldots\]
Sequent calculus

\[
\frac{\Delta, \varphi \vdash \psi}{\Delta \vdash \varphi \rightarrow \psi}
\]

\[
\frac{\Delta ; \varphi \vdash \psi}{\Delta \vdash \varphi \rightarrow \psi}
\]
• Sequent calculus for BI externalizes $\land$ and $\ast$ as different connectives: $;$ and $\cdot$. Only $;$ admits weakening and contraction.
• Sequent calculus for BI externalizes $\land$ and $\ast$ as different connectives: $;$ and $\cdot$. Only $;$ admits weakening and contraction.

• Because of that, contexts in the sequents are not lists/multisets, but trees (referred to as bunches);
• Sequent calculus for BI externalizes $\land$ and $\ast$ as different connectives: $;$ and $\cdot$. Only $;$ admits weakening and contraction.
• Because of that, contexts in the sequents are not lists/multisets, but trees (referred to as *bunches*);
• Left rules can be applied deep inside an arbitrary *bunched context*. 
Cut rule

\[ \Delta' \vdash \psi \quad \Delta(\psi) \vdash \varphi \]

\[ \Delta(\Delta') \vdash \varphi \]
Cut rule

\[
\text{CUT} \quad \frac{\Delta' \vdash \psi \quad \Delta(\psi) \vdash \varphi}{\Delta(\Delta') \vdash \varphi}
\]

Intuitions:

• \( \psi \) is an “intermediate lemma”
Intuitions:

- $\psi$ is an “intermediate lemma”
- provability relation is transitive
Theorem

Everything that is provable, is also provable without the cut rule: $\vdash \varphi \iff \vdash_{\text{cf}} \varphi$
Cut elimination

Theorem
Everything that is provable, is also provable without the cut rule: \( \vdash \varphi \implies \vdash_{\text{cf}} \varphi \)

Why eliminate cut?

- makes the calculus *analytical* (subformula property): any derivation of \( \varphi \vdash \psi \) only involves formula that are already present in \( \varphi \) and \( \psi \)
- important ingredient in the automated proof search toolbox
Cut elimination

Usually proofs of cut elimination involve analysis by inversion + terminating measure:

\[
\begin{align*}
\Delta_1 \vdash \psi_1 \land \psi_2 & \quad \Delta(\psi_1 \land \psi_2) \vdash \varphi \\
\Delta_1 ; \Delta_2 & \vdash \varphi
\end{align*}
\]
Usually proofs of cut elimination involve analysis by inversion + terminating measure:

\[
\begin{align*}
\Delta_1 ; \Delta_2 \vdash \psi_1 \land \psi_2 \\
\Delta(\psi_1 \land \psi_2) \vdash \phi \\
\Delta(\Delta_1 ; \Delta_2) \vdash \phi
\end{align*}
\]
Usually proofs of cut elimination involve analysis by inversion + terminating measure:

\[
\begin{align*}
\Delta_1 \vdash \psi_1 & \quad \Delta_2 \vdash \psi_2 \\
\Delta_1 \mathbin{;} \Delta_2 \vdash \psi_1 \land \psi_2 & \quad \Delta(\psi_1 \mathbin{;} \psi_2) \vdash \varphi \\
\Delta(\Delta_1 \mathbin{;} \Delta_2) \vdash \varphi
\end{align*}
\]
Cut elimination

Usually proofs of cut elimination involve analysis by inversion + terminating measure:

\[ \Delta_1 \vdash \psi_1 \quad \Delta_2 \vdash \psi_2 \]
\[ \Delta_1 ; \Delta_2 \vdash \psi_1 \land \psi_2 \]
\[ \Delta(\psi_1 ; \psi_2) \vdash \phi \]
\[ \Delta(\Delta_1 ; \Delta_2) \vdash \phi \]
Usually proofs of cut elimination involve analysis by inversion + terminating measure:

\[
\frac{\Delta_1 \vdash \psi_1 \quad \Delta_2 \vdash \psi_2}{\Delta_1 ; \Delta_2 \vdash \psi_1 \land \psi_2} \quad \frac{\Delta_1 ; \Delta_2 \vdash \psi_1 \land \psi_2}{\Delta(\psi_1 \land \psi_2) \vdash \varphi} \quad \frac{\Delta(\Delta_1 ; \Delta_2) \vdash \varphi}{\Delta(\Delta_1 ; \Delta_2) \vdash \varphi}
\]
Usually proofs of cut elimination involve analysis by inversion + terminating measure:

\[
\begin{align*}
\Delta_1 \vdash \psi_1 & \quad \Delta_2 \vdash \psi_2 & \Delta(\psi_1 \land \psi_2) ; \varphi_1 \vdash \varphi_2 \\
\Delta_1 ; \Delta_2 \vdash \psi_1 \land \psi_2 & \quad \Delta(\psi_1 \land \psi_2) \vdash \varphi_1 \rightarrow \varphi_2 \\
\Delta(\Delta_1 ; \Delta_2) \vdash \varphi_1 \rightarrow \varphi_2
\end{align*}
\]
Usually proofs of cut elimination involve analysis by inversion + terminating measure:

\[
\begin{align*}
\Delta_1 &\vdash \psi_1 & \Delta_2 &\vdash \psi_2 & \Delta(\psi_1 \land \psi_2) ; \varphi_1 &\vdash \varphi_2 \\
\Delta_1 ; \Delta_2 &\vdash \psi_1 \land \psi_2 & \Delta(\psi_1 \land \psi_2) &\vdash \varphi_1 \rightarrow \varphi_2 & \Delta(\Delta_1 ; \Delta_2) &\vdash \varphi_1 \rightarrow \varphi_2
\end{align*}
\]
Usually proofs of cut elimination involve analysis by inversion + terminating measure:

\[ \Delta_1 ; \Delta_2 \vdash \psi_1 \land \psi_2 \] \[ \Delta(\psi_1 \land \psi_2) \vdash \varphi \]

\[ \Delta(\Delta_1 ; \Delta_2) \vdash \varphi \]

\[ \Rightarrow \]

etc..
Limitations of the direct-style proof

- There are a lot of cases to consider, with a lot of syntactic details
Limitations of the direct-style proof

- There are a lot of cases to consider, with a lot of syntactic details
- Well-foundedness/termination measures can get complicated
Limitations of the direct-style proof

- There are a lot of cases to consider, with a lot of syntactic details
- Well-foundedness/termination measures can get complicated
- BI specific: the tree-like structure of bunches contribute to the complexity
Limitations of the direct-style proof

- There are *a lot* of cases to consider, with a lot of syntactic details
- Well-foundedness/termination measures can get complicated
- BI specific: the tree-like structure of bunches contribute to the complexity

For these reason, non-formalized proofs of cut elimination can be fragile and are known to be error-prone.
Limitations of the direct-style proof

- There are a lot of cases to consider, with a lot of syntactic details
- Well-foundedness/termination measures can get complicated
- BI specific: the tree-like structure of bunches contribute to the complexity

For these reason, non-formalized proofs of cut elimination can be fragile and are known to be error-prone.

On the other hand, formalizing these kind of proofs can also be tough...
A semantic proof of cut elimination goes through some “universal” model $C$ and the interpretation of sequent calculus in it.

$$C \models \varphi \implies \Gamma_{cf} \varphi$$
A semantic proof of cut elimination goes through some “universal” model $C$ and the interpretation of sequent calculus in it.

$$C \models \varphi \implies \Gamma_{cf} \varphi$$

**BI algebra**

A BI algebra $(C, \leq)$ consists of operations $\top, \bot, \lor, \land, \rightarrow, \text{Emp}, \ast, \ast$ satisfying various laws.

**Soundness:** $\varphi \vdash \psi \implies [\varphi] \leq [\psi]$. 
Define $[\varphi] = \{\psi \mid \varphi \vdash \psi\}$, and $[\varphi] \leq_{L} [\psi] \iff \varphi \vdash \psi$. 
Define $[\varphi] = \{ \psi \mid \varphi \vdash \psi \}$, and $[\varphi] \leq_L [\psi] \iff \varphi \vdash \psi$.

- $L = \{ [\varphi] \mid \varphi \in Frml \}$ with $\leq_L$ is a BI algebra;
Intuition: Lindenbaum-Tarski algebra for completeness

Define $[\varphi] = \{\psi \mid \varphi \vdash \psi\}$, and $[\varphi] \leq L [\psi] \iff \varphi \vdash \psi$.

- $L = \{[\varphi] \mid \varphi \in Frml\}$ with $\leq_L$ is a BI algebra;
- Completeness: suppose $\varphi \models \psi$.

The "real" work is to show that $L$ is indeed a model.
Define $[\varphi] = \{ \psi \mid \varphi \vdash \psi \}$, and $[\varphi] \leq L[\psi] \iff \varphi \vdash \psi$.

- $\mathcal{L} = \{ [\varphi] \mid \varphi \in Frml \}$ with $\leq \mathcal{L}$ is a BI algebra;
- Main property of $\mathcal{L}$: $[\varphi] = [\varphi]$.
- Completeness: suppose $\varphi \models \psi$.
  - In particular: $[\varphi] \leq \mathcal{L} [\psi]$, i.e. $[\varphi] \leq L[\psi]$;
Intuition: Lindenbaum-Tarski algebra for completeness

Define $[\varphi] = \{\psi \mid \varphi \vdash \psi\}$, and $[\varphi] \leq_L [\psi] \iff \varphi \vdash \psi$.

- $L = \{[\varphi] \mid \varphi \in Frml\}$ with $\leq_L$ is a BI algebra;
- Completeness: suppose $\varphi \models \psi$.
  - In particular: $[\varphi] \leq_L [\psi]$, i.e. $[\varphi] \leq_L [\psi]$;
  - Conclusion: $\varphi \vdash \psi$. 

The “real” work is to show that $L$ is indeed a model.
Define \([ \varphi ] = \{ \psi \mid \varphi \vdash \psi \}\), and \([ \varphi ] \leq_{L} [ \psi ] \iff \varphi \vdash \psi\).

- \( L = \{ [\varphi] \mid \varphi \in \text{Frml} \}\) with \( \leq_{L} \) is a BI algebra;
- Main property of \( L \): \( [\varphi] = [\varphi] \).
- Completeness: suppose \( \varphi \models \psi \).
  - In particular: \( [\varphi] \leq_{L} [\psi] \), i.e. \( [\varphi] \leq_{L} [\psi] \);
  - Conclusion: \( \varphi \vdash \psi \).
- The “real” work is to show that \( L \) is indeed a model.
Attempt 1

What if we use $\vdash_{cf}$ instead of $\vdash$ in the definition of $\mathcal{L}$?
What if we use $\vdash_{cf}$ instead of $\vdash$ in the definition of $\mathcal{L}$?

Need transitivity of $\leq$: $[\varphi] \leq [\psi] \leq [\chi] \implies [\varphi] \leq [\chi]$?
What if we use $\vdash_{cf}$ instead of $\vdash$ in the definition of $\mathcal{L}$?

Need transitivity of $\leq$: $[\varphi] \leq [\psi] \leq [\chi] \implies [\varphi] \leq [\chi]$?

Same as cut elimination: $\varphi \vdash_{cf} \psi \vdash_{cf} \chi \implies \varphi \vdash_{cf} \chi$
Attempted solution: use sets of predecessors.

\[ \langle \varphi \rangle = \{ \Delta \mid \Delta \vdash_{cf} \varphi \} \in \wp(Bunch), \]

with the subset inclusion relation.
Attempted solution: use sets of predecessors.

$$\langle \varphi \rangle = \{ \Delta \mid \Delta \vdash_{cf} \varphi \} \in \wp(\text{Bunch}),$$

with the subset inclusion relation.

Note that $\varphi \in \langle \varphi \rangle$. Hence, $\langle \varphi \rangle \subseteq \langle \psi \rangle$ implies

$$\varphi \in \langle \psi \rangle \iff \varphi \vdash_{cf} \psi.$$
Attempted solution: use sets of predecessors.

\[ \varphi \vdash_{cf} \varphi \langle \varphi \rangle = \{ \Delta \mid \Delta \vdash_{cf} \varphi \} \subseteq \mathcal{P}(\mathcal{Bunch}), \]

with the subset inclusion relation.

Note that \( \varphi \in \langle \varphi \rangle \). Hence, \( \langle \varphi \rangle \subseteq \langle \psi \rangle \) implies

\[ \varphi \in \langle \psi \rangle \iff \varphi \vdash_{cf} \psi. \]
Is $C = (\{ \langle \varphi \rangle \mid \varphi \in Frml \}, \subseteq)$ a BI algebra?
Is $C = \{\langle \varphi \rangle \mid \varphi \in Frml \}, \subseteq$ a BI algebra?

Not closed under $\cup$, $\cap$... Cannot inherit the algebra structure from $\wp(Bunch)$.

For example, $(\varphi \lor \psi) \in \langle \varphi \lor \psi \rangle$, but does $(\varphi \lor \psi)$ belong to $\langle \varphi \rangle \cup \langle \psi \rangle$?
Is $C = \{\langle \varphi \rangle \mid \varphi \in Frml\}, \subseteq$ a BI algebra?

**Not closed under $\cup$, $\cap$...** Cannot inherit the algebra structure from $\mathcal{B}(Bunch)$.

For example, $(\varphi \lor \psi) \in \langle \varphi \lor \psi \rangle$, but does $(\varphi \lor \psi)$ belong to $\langle \varphi \rangle \cup \langle \psi \rangle$?

Solution: “the next best thing”

$$\langle \varphi \rangle \lor \langle \psi \rangle = \bigcap \{Y \in C \mid \langle \varphi \rangle \cup \langle \psi \rangle \subseteq Y\}$$

The smallest set in $C$ containing $\langle \varphi \rangle \cup \langle \psi \rangle$
Solution: close under arbitrary intersections:

\[ C = \left\{ \bigcap_{i \in I} \langle \varphi_i \rangle \mid I \text{ arbitrary set, } \varphi_i \in Frml \right\} \subseteq \wp(Bunch) \]

\[ \text{cl}(-) : \wp(Bunch) \to C \]

\[ \text{cl}(X) = \bigcap\{ \langle \varphi \rangle \mid X \subseteq \langle \varphi \rangle \} \]
Attempt 3.5 (successful and final)

Solution: close under arbitrary intersections:

\[ C = \left\{ \bigcap_{i \in I} \langle \varphi_i \rangle \mid I \text{ arbitrary set, } \varphi_i \in \text{Frml} \right\} \subseteq \wp(\text{Bunch}) \]

The smallest set in \( C \) containing \( X \)

\[ \text{cl}(-) : \wp(\text{Bunch}) \to C \]

\[ \text{cl}(X) = \bigcap \{ \langle \varphi \rangle \mid X \subseteq \langle \varphi \rangle \} \]
Solution: close under arbitrary intersections:

\[ \mathcal{C} = \{ \bigcap_{i \in I} \langle \varphi_i \rangle \mid I \text{ arbitrary set, } \varphi_i \in Frml \} \subseteq \wp(Bunch) \]

\[
\text{cl}(\cdot) : \wp(Bunch) \rightarrow \mathcal{C}
\]

\[
\text{cl}(X) = \bigcap \{ \langle \varphi \rangle \mid X \subseteq \langle \varphi \rangle \}
\]

**Proposition**

If \( X \in \mathcal{C} \) with \( \Delta_1, \ldots, \Delta_n \in X \) and

\[
\Delta_1 \vdash \varphi \quad \ldots \quad \Delta_n \vdash \varphi
\]

without the use of the cut rule, then \( \Delta' \in X \).
Attempt 3.5 (successful and final)

Solution: close under arbitrary intersections:

\[ C = \{ \bigcap_{i \in I} \langle \varphi_i \rangle \mid I \text{ arbitrary set}, \varphi_i \in Frml \} \subseteq \wp(Bunch) \]

\[ cl(-) : \wp(Bunch) \to C \]

\[ cl(X) = \bigcap \{ \langle \varphi \rangle \mid X \subseteq \langle \varphi \rangle \} \]

Lift operations to \( C \):

\[ X \land Y = X \cap Y \]

\[ X \lor Y = cl(X \cup Y) \]

\[ X \ast Y = cl(\{ \Delta_1 \ast \Delta_2 \mid \Delta_1 \in X, \Delta_2 \in Y \}) \]

\[ X \rightarrow Y = \{ \Delta \mid \forall \Delta' \in X. (\Delta ; \Delta') \in Y \} \]

\[ X \rightarrow \ast Y = \{ \Delta \mid \forall \Delta' \in X. (\Delta \ast \Delta') \in Y \} \]

**Proposition**

\( C \) is a BI algebra
Fundamental property

$\varphi \in \llbracket \varphi \rrbracket \subseteq \langle \varphi \rangle$

Proof by induction on $\varphi$. 
Fundamental property

\[ \varphi \in [\varphi] \subseteq \langle \varphi \rangle \]

Proof by induction on \( \varphi \).

Cut elimination

\[ \varphi \vdash \psi \implies \varphi \vdash_{\text{cf}} \psi \]

If \( \varphi \vdash \psi \), then \([\varphi] \subseteq [\psi] \).

If \([\varphi] \subseteq [\psi] \), then \( \varphi \in [\varphi] \subseteq [\psi] \subseteq \langle \psi \rangle \implies \varphi \vdash_{\text{cf}} \psi \).
The semantic approach is, arguably, more extensible.
The semantic approach is, arguably, more extensible.

The key points in the proof:

• Invertibility of certain rules w.r.t. $\vdash_{cf}$. 
The semantic approach is, arguably, more extensible.

The key points in the proof:

- Invertibility of certain rules w.r.t. $\vdash_{\text{cf}}$.
- The resulting $C$ is a BI algebra.
The semantic approach is, arguably, more extensible.

The key points in the proof:

- Invertibility of certain rules w.r.t. $\because_{cf}$.
- The resulting $\mathcal{C}$ is a BI algebra
- $\varphi \in \llbracket \varphi \rrbracket \subseteq \langle \varphi \rangle$
We consider two different types of extensions:

- BI + additional structural rules, e.g. affine BI
  \[ \Pi(\Delta) \vdash \phi \]
  \[ \Pi(\Delta', \Delta) \vdash \phi \]
- BI + □ modality: BI based on IS4
  \[ \boxed{\Delta(A)} \vdash B \]
  \[ \boxed{\Delta(\boxed{A})} \vdash B \]
  \[ \boxed{\Delta} \vdash A \]
  \[ \boxed{\Delta} \vdash \boxed{A} \]
We consider two different types of extensions:

- **BI + additional structural rules, e.g. affine BI**

  \[
  \Pi(\Delta) \vdash \varphi \\
  \Pi(\Delta, \Delta') \vdash \varphi
  \]

  \(C\) is a BI algebra + some equations, e.g. \(p \ast q \leq p\)
We consider two different types of extensions:

- BI + additional structural rules, e.g. affine BI

\[
\begin{align*}
\Pi(\Delta) & \vdash \varphi \\
\Pi(\Delta, \Delta') & \vdash \varphi
\end{align*}
\]

\(C\) is a BI algebra + some equations, e.g. \(p \ast q \leq p\)
We consider two different types of extensions:

- **BI + additional structural rules, e.g. affine BI**

\[
\begin{align*}
\Pi(\Delta) & \vdash \varphi \\
\Pi(\Delta, \Delta') & \vdash \varphi
\end{align*}
\]

\(C\) is a BI algebra + some equations, e.g. \(p \star q \leq p\)

- **BI + □ modality: BI based on IS4**

\[
\begin{align*}
\square L & \\
\Delta(A) & \vdash B \\
\Delta(\square A) & \vdash B
\end{align*}
\]

\[
\begin{align*}
\square R & \\
\square \Delta & \vdash A \\
\square \Delta & \vdash \square A
\end{align*}
\]

\(C\) is a BI algebra with a modal operator
Extensions (analytic structural rules)

An analytic structural rule is of the form

$$\Pi(T_1[\Delta_1, \ldots, \Delta_n]) \vdash \varphi \quad \ldots \quad \Pi(T_m[\Delta_1, \ldots, \Delta_n]) \vdash \varphi$$

$$\Pi(T[\Delta_1, \ldots, \Delta_n]) \vdash \varphi$$

where $T_1, \ldots, T_m, T$ are *bunched terms* – bunches built out of connectives $\&$, $\mid$, and variables $x_1, \ldots, x_n$, and $T$ is *linear*
An analytic structural rule is of the form

\[
\Pi(T_1[\Delta_1, \ldots, \Delta_n]) \vdash \varphi \quad \ldots \quad \Pi(T_m[\Delta_1, \ldots, \Delta_n]) \vdash \varphi
\]

\[
\Pi(T[\Delta_1, \ldots, \Delta_n]) \vdash \varphi
\]

where \(T_1, \ldots, T_m, T\) are \textit{bunched terms} – bunches built out of connectives \&, \,, and variables \(x_1, \ldots, x_n\), and \(T\) is \textit{linear}

Corresponds to the axiom:

\[
[T](p_1, \ldots, p_n) \leq [T_1](p_1, \ldots, p_n) \lor \cdots \lor [T_m](p_1, \ldots, p_n).
\]
How do we verify that

\[ [T](p_1, \ldots, p_n) \leq [T_1](p_1, \ldots, p_n) \lor \cdots \lor [T_m](p_1, \ldots, p_n). \]

holds in the model \( C \)?

**Proposition**

For \( X_1, X_2, \ldots, X_n \in C \),

\[ cl(\{T[\Delta_1, \ldots, \Delta_n] \mid \Delta_i \in X_i, 1 \leq i \leq n\}) \subseteq [T](X_1, \ldots, X_n). \]

And this becomes an equality when \( T \) is linear.
Analytic completion

\[ \Pi(T_1[\Delta_1, \ldots, \Delta_n]) \vdash \varphi \quad \ldots \quad \Pi(T_m[\Delta_1, \ldots, \Delta_n]) \vdash \varphi \]

\[ \Pi(T[\Delta_1, \ldots, \Delta_n]) \vdash \varphi \]

What if \( T \) is not linear?
Analytic completion

$$\Pi(T_1[\Delta_1, \ldots, \Delta_n]) \vdash \varphi \quad \ldots \quad \Pi(T_m[\Delta_1, \ldots, \Delta_n]) \vdash \varphi$$

$$\Pi(T[\Delta_1, \ldots, \Delta_n]) \vdash \varphi$$

What if $T$ is not linear?

Then we can turn the above rule into an equivalent analytic rule.
Analytic completion

\[ \Pi(\Delta) \vdash \varphi \]

\[ \therefore \Pi(\Delta \cup \Delta) \vdash \varphi \]

corresponds to \( p \ast p \leq p \)
Analytic completion

\[ \Pi(\Delta) \vdash \varphi \]
\[ \Pi(\Delta, \Delta) \vdash \varphi \]

corresponds to \( p \ast p \leq p \)

\[ \Rightarrow \]

\[ \Pi(\Delta_1, \Delta_2) \vdash \varphi \]
Analytic completion

\[
\frac{\Pi(\Delta) \vdash \varphi}{\Pi(\Delta, \Delta) \vdash \varphi}
\]

corresponds to \( p \ast p \leq p \)

\[
\Rightarrow
\]

\[
\frac{\Pi(\Delta_1) \vdash \varphi \quad \Pi(\Delta_2) \vdash \varphi}{\Pi(\Delta_1, \Delta_2) \vdash \varphi}
\]

corresponds to \( p_1 \ast p_2 \leq p_1 \lor p_2 \)
Clearly $p_1 * p_2 \leq p_1 \lor p_2$ implies $p * p \leq p$.

For the other way around:

$$p_1 * p_2 \leq \ldots$$
Clearly \( p_1 \ast p_2 \leq p_1 \lor p_2 \) implies \( p \ast p \leq p \).

For the other way around:

\[
p_1 \ast p_2 \leq (p_1 \lor p_2) \ast (p_1 \lor p_2) \leq \]

Clearly $p_1 * p_2 \leq p_1 \lor p_2$ implies $p * p \leq p$.

For the other way around:

$$p_1 * p_2 \leq (p_1 \lor p_2) * (p_1 \lor p_2) \leq p_1 \lor p_2.$$
Reflecting on the formalization

Coq formalization, ~1090 lines specs and ~3600 lines proof

Record C := {
  CPred :> Bunch → Prop;
  CClosed : .... }

• Extensive use of setoids and setoid rewriting, based on the typeclasses from the stdpp library

• Turn equations $\Delta = \Delta'$ into inductive systems

Inductive bunch_decomp : bunch → bunch_ctx → bunch → Prop
Reflecting on the formalization

Coq formalization, ~1090 lines specs and ~3600 lines proof

• Good representation for $C$ makes life easier

Record $C := \{$

\begin{align*}
  \text{CPred} & : \rightarrow \text{Bunch} \rightarrow \text{Prop}; \\
  \text{CClosed} & : \ldots.
\end{align*}

}
Reflecting on the formalization

Coq formalization, ~1090 lines specs and ~3600 lines proof

• Good representation for $C$ makes life easier

```coq
Record C := {
  CPred :> Bunch → Prop;
  CClosed : .... }
```

• Extensive use of setoids and setoid rewriting, based on the typeclasses from the stdpp library
Reflecting on the formalization

Coq formalization, ~1090 lines specs and ~3600 lines proof

- Good representation for $C$ makes life easier

  Record $C := \{$
  
  CPred :> Bunch $\rightarrow$ Prop;
  CClosed : .... $

- Extensive use of setoids and setoid rewriting, based on the typeclasses from the stdpp library

- Turn equations $\Delta = \Delta'(\Gamma)$ into inductive systems

  Inductive bunch_decomp : bunch $\rightarrow$ bunch_ctx $\rightarrow$ bunch $\rightarrow$ Prop
Thank you for your attention!
Let me know if you have questions, d.frumin@rug.nl.