Retractions in Multiplicative Linear Logic

Rémi Di Guardia, Olivier Laurent

ENS Lyon (LIP)

Chocola 13/03/2024
**Isomorphisms**

Isomorphisms relate types/formulas/objects which are “the same”

\[ A \simeq B \]

\[ id_A \quad f \quad g \quad id_B \]

Instantiation in $\lambda$-calculus, logics, …

Wanted: an *equational theory*

Two main approaches:

**Syntactic**  
the analysis of pairs of terms composing to the identity  
should provide information on their type

**Semantic**  
find a model with the same isomorphisms than in the syntax  
but where they can be computed more easily (typically  
reducing to equality between combinatorial objects)
Isomorphisms relate types/formulas/objects which are “the same”

\[ A \simeq B \]

For \( \lambda \)-calculus with products and unit type / cartesian closed categories

Semantic (finite sets) [Soloviev, 1983]

<table>
<thead>
<tr>
<th>( \times )</th>
<th>( A \times (B \times C) \simeq (A \times B) \times C )</th>
<th>( A \times B \simeq B \times A )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \times ) and ( \rightarrow )</td>
<td>( (A \times B) \rightarrow C \simeq A \rightarrow (B \rightarrow C) )</td>
<td>( A \rightarrow (B \times C) \simeq (A \rightarrow B) \times (A \rightarrow C) )</td>
</tr>
<tr>
<td>1</td>
<td>( A \times 1 \simeq A )</td>
<td>( 1 \rightarrow A \simeq A )</td>
</tr>
</tbody>
</table>

Reduces to Tarski’s High School Algebra Problem: can all equalities involving product, exponential and 1 be found using only

\[
\begin{align*}
    a(bc) &= (ab)c \\
    ab &= ba \\
    c^{ab} &= (c^b)^a \\
    (bc)^a &= b^a c^a \\
    1a &= a \\
    a^1 &= a \\
    1^a &= 1
\end{align*}
\]

Rémi Di Guardia, Olivier Laurent
Isomorphisms relate types/formulas/objects which are “the same”

\[ A \simeq B \]

For Multiplicative Linear Logic / ⋆-autonomous categories

Syntactic (proof-nets) [Balat and Di Cosmo, 1999]

\begin{align*}
\text{Associativity} & : & A \otimes (B \otimes C) & \simeq (A \otimes B) \otimes C & & A \lhd (B \lhd C) & \simeq (A \lhd B) \lhd C \\
\text{Commutativity} & : & A \otimes B & \simeq B \otimes A & & A \lhd B & \simeq B \lhd A \\
\text{Neutrality} & : & A \otimes 1 & \simeq A & & A \lhd \bot & \simeq A \\
\end{align*}

\[(A \otimes B) \rightarrow C = (A \lhd \bot \; B \lhd \bot) \lhd C \simeq A \lhd \bot \; (B \lhd \bot \; C) = A \rightarrow (B \rightarrow C)\]
Isomorphisms relate types/formulas/objects which are “the same”

\[ A \simeq B \]

\[ \text{id}_A \quad f \quad g \quad \text{id}_B \]

For Multiplicative-Additive Linear Logic / *-autonomous categories with finite products

<table>
<thead>
<tr>
<th></th>
<th>Associativity</th>
<th>Distributivity</th>
<th>Neutrality</th>
<th>Commutativity</th>
<th>Annihilation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C )</td>
<td>( A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C )</td>
<td>( A \otimes 1 \simeq A )</td>
<td>( A \otimes (B \otimes C) \simeq (A \otimes B) \otimes (A \otimes C) )</td>
<td>( A \otimes 0 \simeq 0 )</td>
<td></td>
</tr>
<tr>
<td>( A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C )</td>
<td>( A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C )</td>
<td>( A \notimes \bot \simeq A )</td>
<td>( A \otimes (B \otimes C) \simeq (A \notimes B) \notimes (A \notimes C) )</td>
<td>( A \notimes \top \simeq \top )</td>
<td></td>
</tr>
<tr>
<td>( A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C )</td>
<td>( A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C )</td>
<td>( A \notimes \bot \simeq A )</td>
<td>( A \otimes (B \otimes C) \simeq (A \notimes B) \notimes (A \notimes C) )</td>
<td>( A \notimes \top \simeq \top )</td>
<td></td>
</tr>
</tbody>
</table>
Isomorphisms
Isomorphisms relate types/formulas/objects which are “the same”

\[ A \simeq B \]

\[ id_A \xleftarrow{f} A \xrightarrow{g} B \xleftarrow{id_B} \]

For Polarized Linear Logic
Semantic (games, forest isomorphisms) [Laurent, 2005]

<table>
<thead>
<tr>
<th>Property</th>
<th>Associativity</th>
<th>Commutativity</th>
<th>Neutrality</th>
<th>Distributivity</th>
<th>Annihilation</th>
<th>Seely</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C ) \hspace{5em} ( A \parr (B \parr C) \simeq (A \parr B) \parr C ) \hspace{5em} ( A \multimap (B \multimap C) \simeq (A \multimap B) \multimap C ) \hspace{5em} ( A &amp; (B &amp; C) \simeq (A &amp; B) &amp; C )</td>
<td>( A \otimes B \simeq B \otimes A ) \hspace{5em} ( A \multimap B \simeq B \multimap A ) \hspace{5em} ( A \oplus B \simeq B \oplus A ) \hspace{5em} ( A &amp; B \simeq B &amp; A )</td>
<td>( A \otimes 1 \simeq A ) \hspace{5em} ( A \multimap \bot \simeq A ) \hspace{5em} ( A \oplus 0 \simeq A ) \hspace{5em} ( A &amp; \top \simeq A )</td>
<td>( A \otimes (B \oplus C) \simeq (A \otimes B) \oplus (A \otimes C) ) \hspace{5em} ( A \multimap (B &amp; C) \simeq (A \multimap B) &amp; (A \multimap C) )</td>
<td>( A \otimes 0 \simeq 0 ) \hspace{5em} ( A \multimap \top \simeq \top )</td>
<td>( !(A &amp; B) \simeq !A \otimes !B ) \hspace{5em} ( ?(A \oplus B) \simeq ?A \multimap ?B ) \hspace{5em} ( !\top \simeq 1 ) \hspace{5em} ( ?0 \simeq \bot )</td>
</tr>
</tbody>
</table>
Retractions relate $A$ and $B$ when $A$ is a “sub-type” of $B$

\[ A \subseteq B \]

\[
\begin{array}{ccc}
\text{id}_A & \xleftarrow{f} & \text{id}_B \\
\text{id}_A & \xrightarrow{g} & \end{array}
\]

Instantiation in $\lambda$-calculus, logics,…

bool $\subseteq$ nat with $f(\text{false}) = 0$, $f(\text{true}) = 1$ and $g(n) = n$ is equal to 1
Retractions relate $A$ and $B$ when $A$ is a “sub-type” of $B$

$$A \trianglelefteq B$$

Instantiation in $\lambda$-calculus, logics,...

```
bool \trianglelefteq \text{nat} with \ f(\text{false}) = 0, \ f(\text{true}) = 42 \text{ and } g(n) = n \text{ is equal to } 42
```
Retractions relate $A$ and $B$ when $A$ is a “sub-type” of $B$

$$A \trianglelefteq B$$

For simply typed affine λ-calculus

Syntactic [Regnier and Urzyczyn, 2002]

<table>
<thead>
<tr>
<th>$\trianglelefteq$</th>
<th>$A \rightarrow B \rightarrow C \trianglelefteq B \rightarrow A \rightarrow C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\triangle$ ($= \triangle \setminus \sim$)</td>
<td>$A \triangle (A \rightarrow X) \rightarrow X$ if $A$ is $Y_1 \rightarrow Y_2 \rightarrow \cdots \rightarrow X$</td>
</tr>
</tbody>
</table>
Retractions relate $A$ and $B$ when $A$ is a “sub-type” of $B$

$$A \triangleleft B$$

For Multiplicative Linear Logic

\[ \simeq \quad \text{associativity and commutativity of } \otimes \text{ and } \& , \text{ neutrality of } 1 \text{ and } \bot \]

<table>
<thead>
<tr>
<th>(\simeq)</th>
<th>(\triangleleft) ((\equiv \triangleleft \setminus \simeq))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\simeq)</td>
<td>(\triangleleft) ((\equiv \triangleleft \setminus \simeq))</td>
</tr>
</tbody>
</table>
Decidability of retractions in simply typed λ-calculus in [Padovani, 2001]

**Definition**

*Cantor-Bernstein* property: if $A \subseteq B$ and $B \subseteq A$ then $A \simeq B$.

Holds in some category but not all!
1. **Multiplicative Linear Logic**
   - Proof-Net
   - Retraction

2. **Properties of Retractions**

3. **Difficulties for the general case** $A \triangleleft B$

4. **Retractions of the shape** $X \triangleleft \cdot$ (universal super-types)
   - Looking for a pattern
   - Quasi-Beffara
   - Beffara $X \triangleleft X \otimes (X \perp \forall X)$
   - Does not generalize to $A \triangleleft B$

5. **Conclusion**
Formula of MLL

Formula

\[ A, B ::= X \mid X^\perp \mid A \otimes B \mid A \pitchfork B \]

Duality

\[
\begin{align*}
(X^\perp)^\perp &= X \\
(A \otimes B)^\perp &= B^\perp \pitchfork A^\perp \\
(A \pitchfork B)^\perp &= B^\perp \otimes A^\perp
\end{align*}
\]
**Formula of MLL**

**Formula**

\[ A, B ::= X \mid X\perp \mid A \otimes B \mid A \oslash B \]

**Duality**

\[
\begin{align*}
(X\perp)\perp &= X \\
(A \otimes B)\perp &= B\perp \oslash A\perp \\
(A \oslash B)\perp &= B\perp \otimes A\perp
\end{align*}
\]

**Sequent**

\[ \vdash A_1, \ldots, A_n \]
**Formula of MLL**

**Formula**

\[ A, B ::= X \mid X^\perp \mid A \otimes B \mid A \parr B \]

**Duality**

\[
(X^\perp)^\perp = X \\
(A \otimes B)^\perp = B^\perp \parr A^\perp \\
(A \parr B)^\perp = B^\perp \otimes A^\perp
\]

**Sequent**

\[ \vdash A_1, \ldots, A_n \]

**Rules (sequent calculus)**

- \[ \vdash A^\perp, A \quad \text{ax} \]
- \[ \vdash A, \Gamma \vdash B, \Delta \quad \otimes \]
- \[ \vdash A, B, \Gamma \quad \parr \]

---

Rémi Di Guardia, Olivier Laurent

Retractions in MLL

Chocola 13/03/2024 6 / 41
Sequent ⊢ A, B with edges between dual leaves (some \(X\) and \(X^\perp\)), these edges partitioning the leaves of the sequent.
Sequent $\vdash A, B$ with edges between dual leaves (some $X$ and $X^\perp$), these edges partitioning the leaves of the sequent.

Examples

Graphical representation
Correctness & Proof-Net

Correctness Graph

In a proof-structure, keep only one premise of each $\otimes$-node.

Danos-Regnier Correctness Criterion

A proof-structure is correct, and called a proof-net, if all its correctness graphs are acyclic and connected (i.e. are trees).

Toy examples

![Toy examples](image.png)
Correctness Graph

In a proof-structure, keep only one premise of each \( \otimes \)-node.

Danos-Regnier Correctness Criterion

A proof-structure is correct, and called a proof-net, if all its correctness graphs are acyclic and connected (i.e., are trees).

Toy examples

- Not acyclic (but connected)
- INCORRECT
Correctness & Proof-Net

Correctness Graph
In a proof-structure, keep only one premise of each \(\Rightarrow\)-node.

Danos-Regnier Correctness Criterion
A proof-structure is correct, and called a proof-net, if all its correctness graphs are acyclic and connected (i.e. are trees).

Toy examples

Not acyclic (but connected)
INCORRECT

Acyclic and connected
**Correctness & Proof-Net**

**Correctness Graph**
In a proof-structure, keep only one premise of each \( \otimes \)-node.

**Danos-Regnier Correctness Criterion**
A proof-structure is correct, and called a proof-net, if all its correctness graphs are acyclic and connected (i.e. are trees).

**Toy examples**
- Not acyclic (but connected)
  - INCORRECT
- Acyclic and connected
  - CORRECT
Correctness & Proof-Net

Correctness Graph

In a proof-structure, keep only one premise of each \( \otimes \)-node.

Danos-Regnier Correctness Criterion

A proof-structure is correct, and called a proof-net, if all its correctness graphs are acyclic and connected (i.e. are trees).

Examples

\[
\begin{align*}
\text{Left: } & Z \otimes X \quad X \quad X \perp \quad Z \perp \\
\text{Center: } & Z \otimes X \quad X \perp \quad Z \perp \\
\text{Right: } & Z \otimes X \quad X \perp \quad Z \perp
\end{align*}
\]
Correctness & Proof-Net

Correctness Graph

In a proof-structure, keep only one premise of each $\otimes$-node.

Danos-Regnier Correctness Criterion

A proof-structure is correct, and called a proof-net, if all its correctness graphs are acyclic and connected (i.e. are trees).

Examples

Acyclic and connected
Correctness Graph

In a proof-structure, keep only one premise of each \( \otimes \)-node.

Danos-Regnier Correctness Criterion

A proof-structure is correct, and called a proof-net, if all its correctness graphs are acyclic and connected (i.e. are trees).

Examples

Not acyclic nor connected

INCORRECT
Correctness & Proof-Net

Correctness Graph
In a proof-structure, keep only one premise of each $\otimes$-node.

Danos-Regnier Correctness Criterion
A proof-structure is correct, and called a proof-net, if all its correctness graphs are acyclic and connected (i.e. are trees).

Examples
Not acyclic nor connected INCORRECT

Acyclic and connected
Correctness & Proof-Net

Correctness Graph
In a proof-structure, keep only one premise of each \( \otimes \)-node.

Danos-Regnier Correctness Criterion
A proof-structure is correct, and called a proof-net, if all its correctness graphs are acyclic and connected (i.e. are trees).

Examples

Not acyclic nor connected
INCORRECT

Acyclic and connected
Correctness & Proof-Net

Correctness Graph

In a proof-structure, keep only one premise of each \( \otimes \)-node.

Danos-Regnier Correctness Criterion

A proof-structure is correct, and called a proof-net, if all its correctness graphs are acyclic and connected (i.e. are trees).

Examples

- Not acyclic nor connected
  
  INCORRECT

- Acyclic and connected
Correctness & Proof-Net

Correctness Graph
In a proof-structure, keep only one premise of each \( \otimes \)-node.

Danos-Regnier Correctness Criterion
A proof-structure is correct, and called a proof-net, if all its correctness graphs are acyclic and connected (i.e. are trees).

Examples
Not acyclic nor connected
INCORRECT

Acyclic and connected
CORRECT
Identity proof-net

Identity proof-structure of $A$

In the sequent $\vdash A^\perp, A$, link each leaf in $A$ to the dual one in $A^\perp$.

Example: $A = Y \otimes (X^\perp \triangledown X^\perp)$

Lemma

An identity proof-structure is correct.
Equivalence Class of a leaf

Take two proof-nets on \( \vdash A, B \) and \( \vdash B^\perp, C \). Forget the syntax trees, keep only the leaves, the axiom edges and put edges between dual leaves of \( B \) and \( B^\perp \).

Equivalence class of a leaf: those connected to it in this graph.
A graph containing only vertices of degree 1 or 2 is a disjoint union of non-empty simple paths and cycles.
A graph containing only vertices of degree 1 or 2 is a disjoint union of non-empty simple paths and cycles.

Thus an equivalence class contains exactly either two leaves of $A$ and $C$ or zero (for they are of degree 1).

Using the correctness criterion, there are no cycles; hence each class contains exactly two leaves of $A$ and $C$. \textit{(But we do not need it here.)}
Composition

Take two proof-structures on \( \vdash A, B \) and \( \vdash B \bot, C \). Delete edges involving leaves of \( B \) and \( B \bot \) and add edges between leaves of \( A \) and \( B \) in the same equivalence class, obtaining a proof-structure on \( \vdash A, C \).

Lemma

*The composition of two proof-nets is a proof-net.*
Composition

Take two proof-structures on $\vdash A, B$ and $\vdash B^\perp, C$. Delete edges involving leaves of $B$ and $B^\perp$ and add edges between leaves of $A$ and $B$ in the same equivalence class, obtaining a proof-structure on $\vdash A, C$.

Lemma

The composition of two proof-nets is a proof-net.

Orthogonality of GOI / of Danos-Regnier

Composition of permutations, yielding a permutation if they are orthogonal

$=$ there are no cycles, only paths

- permutation
- permutation
Example of composition
Example of composition
Example of composition
Example of composition
Example of composition
Example of composition
Example of composition
Retraction

Category theory

\[ \text{id}_A \quad f \quad g \]

\[ A \xleftarrow{\triangle} B \]

\[ \lambda \text{-calculus} \]

Retraction \( A \triangleleft B \)

Terms \( M : A \rightarrow B \) and \( N : B \rightarrow A \) such that

\[ N \circ M =_{\beta\eta} \lambda x^A. x \]
Retraction

Category theory

\[ A \triangleleft B \]

\[ \begin{array}{ccc}
  id_A & \circ & f \\
  \downarrow & & \downarrow \\
  A & \circ & g \\
  \downarrow & & \downarrow \\
  B & \circ & B
\end{array} \]

\[ \lambda \text{-calculus} \]

**Retraction** \( A \triangleleft B \)

Terms \( M : A \to B \) and \( N : B \to A \) such that

\[ N \circ M =_{\beta \eta} \lambda x^A.x \]

**Multiplicative Linear Logic**

**Retraction** \( A \triangleleft B \)

Proof-nets \( \mathcal{R} \) of \( \vdash A^\perp, B \) and \( \mathcal{S} \) of \( \vdash B^\perp, A \) whose composition over \( B \) yields the identity proof-net of \( A \).
Retraction

Category theory

\[
\begin{aligned}
    \text{id}_A & \quad \quad \quad \quad f \\
    A & \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad B
\end{aligned}
\]

\[\lambda\text{-calculus}\]

Retraction \(A \triangleleft B\)

Terms \(M : A \to B\) and \(N : B \to A\) such that

\[N \circ M =_{\beta\eta} \lambda x^A.x\]

Multiplicative Linear Logic

Retraction \(A \triangleleft B\)

Proof-nets \(\mathcal{R}\) of \(\vdash A^\perp, B\) and \(\mathcal{S}\) of \(\vdash B^\perp, A\) whose composition over \(B\) yields the identity proof-net of \(A\).

\[A \triangleleft B \iff A^\perp \triangleleft B^\perp\]
Beffara’s retraction

\[ X \triangleleft X \bowtie (X^\perp \otimes X) \quad \text{or dually} \quad X \triangleleft X \otimes (X^\perp \bowtie X) \]

Can also be seen as \( X \triangleleft (X \rightarrow X) \rightarrow X \)

\[ X \bowtie (X^\perp \otimes X) \quad (X^\perp \bowtie X) \otimes X^\perp \]
Beffara’s is a retraction
Beffara’s is a retraction
Beffaara’s is a retraction
Plan

1 Multiplicative Linear Logic
   - Proof-Net
   - Retraction

2 Properties of Retractions

3 Difficulties for the general case $A \triangleleft B$

4 Retractions of the shape $X \triangleleft \cdot$ (universal super-types)
   - Looking for a pattern
   - Quasi-Beffara
   - Beffara $X \triangleleft X \otimes (X \perp \otimes X)$
   - Does not generalize to $A \triangleleft B$

5 Conclusion
**Definition**

A proof-net on \( \vdash A, B \) is *half-bipartite* in \( A \) if there is no link between leaves of \( A \).

**Example**

Half-bipartite in \( X \otimes X \bot \) but not in \( X \otimes (X \otimes (X \otimes X \bot)) \).
Retractions are half-bipartite

Lemma

Proof-nets of $A \triangleleft B$ are half-bipartite in $A^\bot$ and $A$ respectively.

Proof.

A link between leaves of $A^\bot$ or $A$ would survive in the composition, i.e. in the resulting identity proof-net: contradiction.
Non-ambiguity

Corollary: Non-ambiguity

Up to renaming leaves, in \( A \triangleleft B \) one can assume \( A \) to be non-ambiguous: its leaves are distinct atoms \( X, Y \bot, Z, \ldots \) without \( X \bot, Y, Z \bot \ldots \)

Proof.

Rename each equivalence class with a fresh atom.
Corollary: Non-ambiguity

*Up to renaming leaves, in $A \trianglelefteq B$ one can assume $A$ to be non-ambiguous: its leaves are distinct atoms $X, Y \perp, Z, \ldots$ without $X \perp, Y, Z \perp \ldots$*

Proof.

Rename each equivalence class with a fresh atom.
Corollary: Non-ambiguity

Up to renaming leaves, in $A \trianglelefteq B$ one can assume $A$ to be non-ambiguous: its leaves are distinct atoms $X, Y^\perp, Z, \ldots$ without $X^\perp, Y, Z^\perp \ldots$

Proof.

Rename each equivalence class with a fresh atom.
Corollary: Non-ambiguity

Up to renaming leaves, in $A \trianglelefteq B$ one can assume $A$ to be non-ambiguous: its leaves are distinct atoms $X, Y \perp, Z, \ldots$ without $X \perp, Y, Z \perp \ldots$

Proof.

Rename each equivalence class with a fresh atom.
Corollary: Non-ambiguity

Up to renaming leaves, in $A \trianglelefteq B$ one can assume $A$ to be non-ambiguous: its leaves are distinct atoms $X, Y \perp, Z, \ldots$ without $X \perp, Y, Z \perp \ldots$

Proof.

Rename each equivalence class with a fresh atom.
Corollary: Non-ambiguity

Up to renaming leaves, in $A \triangleleft B$ one can assume $A$ to be non-ambiguous: its leaves are distinct atoms $X, Y \perp, Z, \ldots$ without $X \perp, Y, Z \perp \ldots$

Proof.

Rename each equivalence class with a fresh atom.

1. Dual leaves of $B$ and $B \perp$ in the same equivalence class $\rightarrow B_2 = B_1 \perp$
Corollary: Non-ambiguity

Up to renaming leaves, in $A \triangleleft B$ one can assume $A$ to be non-ambiguous: its leaves are distinct atoms $X, Y^\perp, Z, \ldots$ without $X^\perp, Y, Z^\perp \ldots$

Proof.

Rename each equivalence class with a fresh atom.

1. Dual leaves of $B$ and $B^\perp$ in the same equivalence class $\rightarrow B_2 = B_1^\perp$
2. Composition is identity $\rightarrow$ dual leaves of $A^\perp$ and $A$ in the same equivalence class $\rightarrow A_2 = A_1^\perp$ non-ambiguous
Corollary: Non-ambiguity

Up to renaming leaves, in $A \trianglelefteq B$ one can assume $A$ to be non-ambiguous: its leaves are distinct atoms $X, Y \perp, Z, \ldots$ without $X \perp, Y, Z \perp \ldots$

Proof.

Rename each equivalence class with a fresh atom.

1. Dual leaves of $B$ and $B \perp$ in the same equivalence class $\rightarrow B_2 = B_1 \perp$
2. Composition is identity $\rightarrow$ dual leaves of $A \perp$ and $A$ in the same equivalence class $\rightarrow A_2 = A_1 \perp$ non-ambiguous
3. Renaming preserves correction and the result of composition
Property on sizes

If $A$ non-ambiguous, there is only one proof-net on $\vdash A^\perp, A$: the identity.

Retraction $A \preceq B$ with $A$ non-ambiguous

Proof-nets $R$ of $\vdash A^\perp, B$ and $S$ of $\vdash B^\perp, A$ whose composition over $B$ yields the identity proof-net of $A$. 
Property on sizes

If $A$ non-ambiguous, there is only one proof-net on $\vdash A^\perp$, $A$: the identity.

**Retraction $A \preceq B$ with $A$ non-ambiguous**

Proof-nets $\mathcal{R}$ of $\vdash A^\perp$, $B$ and $\mathcal{S}$ of $\vdash B^\perp$, $A$ whose composition over $B$ yields the identity proof-net of $A$.

**Theorem**

*If $A \preceq B$, then $s(A) \leq s(B)$, with equality iff $A \simeq B$.***

**Proof.**

If $s(A) = s(B)$, then each atom of $B$ corresponds to one in $A^\perp$, so $B$ non-ambiguous too. Thus, both compositions yield identities. Reciprocally, associativity and commutativity preserve the size.
Consequences

The previous result on non-ambiguity permits to characterize isomorphisms as done in [Balat and Di Cosmo, 1999]:

<table>
<thead>
<tr>
<th>Associativity</th>
<th>( A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C )</th>
<th>( A \ast (B \ast C) \simeq (A \ast B) \ast C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commutativity</td>
<td>( A \otimes B \simeq B \otimes A )</td>
<td>( A \ast B \simeq B \ast A )</td>
</tr>
</tbody>
</table>
Consequences

The previous result on non-ambiguity permits to characterize isomorphisms as done in [Balat and Di Cosmo, 1999]:

<table>
<thead>
<tr>
<th>Associativity</th>
<th>$A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$</th>
<th>$A \bowtie (B \bowtie C) \simeq (A \bowtie B) \bowtie C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commutativity</td>
<td>$A \otimes B \simeq B \otimes A$</td>
<td>$A \bowtie B \simeq B \bowtie A$</td>
</tr>
</tbody>
</table>

Corollary

*The Cantor-Bernstein property holds:*

$$A \preceq B \text{ and } B \preceq A \implies A \simeq B$$
Consequences

The previous result on non-ambiguity permits to characterize isomorphisms as done in [Balat and Di Cosmo, 1999]:

<table>
<thead>
<tr>
<th>Associativity</th>
<th>$A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$</th>
<th>$A \bowtie (B \bowtie C) \simeq (A \bowtie B) \bowtie C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commutativity</td>
<td>$A \otimes B \simeq B \otimes A$</td>
<td>$A \bowtie B \simeq B \bowtie A$</td>
</tr>
</tbody>
</table>

Corollary

*The Cantor-Bernstein property holds:*

$$A \preceq B \text{ and } B \preceq A \implies A \simeq B$$

$$X \otimes Y \not\preceq X \bowtie Y$$

$$X \bowtie (Y \otimes Z) \not\preceq Y \otimes (X \bowtie Z)$$
Plan

1. **Multiplicative Linear Logic**
   - Proof-Net
   - Retraction

2. **Properties of Retractions**

3. **Difficulties for the general case** $A \subseteq B$

4. **Retractions of the shape** $X \triangleleft \bullet$ (universal super-types)
   - Looking for a pattern
   - Quasi-Beffara
   - Beffara $X \triangleleft X \otimes (X^\bot \bowtie X)$
   - Does not generalize to $A \subseteq B$

5. **Conclusion**
\[ X_1 \otimes X_2 \otimes X_3 \otimes X_4 \not\prec (X_1 \otimes X_2 \otimes X_3 \otimes X_4) \Rightarrow (X_1 \otimes (X_1^\bot \Rightarrow)
(X_2 \otimes (X_2^\bot \Rightarrow)
(X_3 \otimes (X_3^\bot \Rightarrow)
(X_4 \otimes (X_4^\bot )))\))\)

Generally:
\[
\{ \otimes X_i \} \not\prec \{ \otimes X_i \} \Rightarrow (X_1 \otimes (X_1^\bot \Rightarrow (\ldots (X_{n-1} \otimes (X_{n-1}^\bot \Rightarrow (X_n \otimes X_n^\bot )) \ldots )))\)

However \((A \otimes X) \Rightarrow B \not\lhd (A \otimes X) \Rightarrow (X \otimes (X^\bot \Rightarrow B))\)
Plan

1. Multiplicative Linear Logic
   - Proof-Net
   - Retraction

2. Properties of Retractions

3. Difficulties for the general case $A \trianglelefteq B$

4. Retractions of the shape $X \trianglelefteq \cdot$ (universal super-types)
   - Looking for a pattern
   - Quasi-Beffara
   - Beffara $X \trianglelefteq X \otimes (X^\perp \varnothing X)$
   - Does not generalize to $A \trianglelefteq B$

5. Conclusion
**Key Result**

**Lemma**

In $X \triangleleft B$ one of the two proof-nets contains:

![Diagram](https://example.com/diagram.png)

**Proof.**

We build a sequence (GOI path) finding such a pattern.
Lemma

In \( X \triangleleft B \) one of the two proof-nets contains:

Proof.

We build a sequence (GOI path) finding such a pattern.
Lemma

In $X \triangleleft B$ one of the two proof-nets contains:

Proof.

We build a sequence (GOI path) finding such a pattern.
**Key Result**

**Lemma**

\[ X \triangleleft B \text{ one of the two proof-nets contains:} \]

**Proof.**

We build a sequence (GOI path) finding such a pattern.
Key Result

Lemma

In $X \triangleleft B$ one of the two proof-nets contains:

Proof.

We build a sequence (GOI path) finding such a pattern.
In $X \triangleleft B$ one of the two proof-nets contains:

Proof.

We build a sequence (GOI path) finding such a pattern.
**Key Result**

**Lemma**

\[\text{In } X \triangleleft B \text{ one of the two proof-nets contains:} \]

\[
\begin{array}{c}
\text{Proof.}\n\end{array}
[\]

\[\text{We build a sequence (GOI path) finding such a pattern.} \]

\[
\begin{array}{c}
\text{\textbullet\hspace{1cm}B\hspace{1cm}B} \text{\textbullet}\n\end{array}
[\]
In $X \triangleleft B$ one of the two proof-nets contains:

Proof.
We build a sequence (GOI path) finding such a pattern.
In $X \triangleleft B$ one of the two proof-nets contains:

**Proof.**

We build a sequence (GOI path) finding such a pattern.
**Key Result**

**Lemma**

In $X \triangleleft B$ one of the two proof-nets contains:

**Proof.**

We build a sequence (GOI path) finding such a pattern.
**Key Result**

**Lemma**

In $X \triangleleft B$ one of the two proof-nets contains:

**Proof.**

We build a sequence (GOI path) finding such a pattern.
In $X \triangleleft B$ one of the two proof-nets contains:

Proof.
We build a sequence (GOI path) finding such a pattern.

**Invariant:** every $X$ of $B$ is above a $\otimes$, and every $X \perp$ above a $\triangledown$.
In $X \triangleleft B$ one of the two proof-nets contains:

Proof.

We build a sequence (GOI path) finding such a pattern. 

**Invariant:** every $X$ of $B$ is above a $\otimes$, and every $X^\bot$ above a $\Rightarrow$. 

![Diagram showing proof-nets](image-url)
Key Result

Lemma

In $X \triangleleft B$ one of the two proof-nets contains:

Proof.

We build a sequence (GOI path) finding such a pattern. 
Invariant: every $X$ of $B$ is above a $\otimes$, and every $X^\perp$ above a $\triangledown$. 

\[X^\perp_1 \quad X_2 \quad X^\perp_3 \quad X_4 \quad \ldots \quad B \quad \triangledown \]

\[X_3 \quad X^\perp_2 \quad \ldots \quad B^\perp \]
**Key Result**

**Lemma**

In $X \triangleleft B$ one of the two proof-nets contains:

![Diagram of a proof-net containing $X \perp$ and $X$](attachment:image.png)

**Proof.**

We build a sequence (GOI path) finding such a pattern.

**Invariant:** every $X$ of $B$ is above a $\otimes$, and every $X \perp$ above a $\triangleright$.

![Diagram of a sequence of proof-nets](attachment:image.png)
Extended pattern

**Lemma**

If $X \perp X$ has a node below it, then this is a $X \perp X \otimes X$.

**Proof.**

The connector below the pattern cannot be a $\otimes$ by connectivity:
Quasi-Beffara is this local transformation on proofs of a retraction $A \triangleleft B$:

By extension, this defines two transformations on a formula $B$ (by duality):
Lemma

If \((R, S)\) are proofs of \(A \subseteq B\) and \((R, S) \xrightarrow{\text{qBeffara}} (R', S')\), then \((R', S')\) are proofs of \(A \subseteq B'\) with \(B \xrightarrow{\text{qBeffara}} B'\).

Proof.

Quasi-Beffara preserves:

- being a proof-structure
**Lemma**

If \((R, S)\) are proofs of \(A \subseteq B\) and \((R, S) \xrightarrow{q\text{Beffara}} (R', S')\), then \((R', S')\) are proofs of \(A \subseteq B'\) with \(B \xrightarrow{q\text{Beffara}} B'\).

**Proof.**

Quasi-Beffara preserves:
- being a proof-structure
- acyclicity of correctness graphs
Lemma

If \((R, S)\) are proofs of \(A \preceq B\) and \((R, S) \xrightarrow{q\text{Beffa}} (R', S')\), then \((R', S')\) are proofs of \(A \preceq B'\) with \(B \xrightarrow{q\text{Beffa}} B'\).

Proof.

Quasi-Beffa preserves:
- being a proof-structure
- acyclicity of correctness graphs
- the number \(|V| + |\mathfrak{F}| - |E|\) of cc. of any correctness graph: it removes 4 vertices, including 1 \(\mathfrak{F}\), and 5 edges

\[
\begin{align*}
&\xrightarrow{\alpha} \times &\xrightarrow{\beta} \times \\
&\times &\times
\end{align*}
\]

\[
\xrightarrow{\text{qBeffa}}
\]

\[
\xrightarrow{\alpha} \quad \xrightarrow{\beta}
\]
Lemma

If \((\mathcal{R}, S)\) are proofs of \(A \trianglelefteq B\) and \((\mathcal{R}, S) \xrightarrow{q\text{Beffa}ra} (\mathcal{R}', S')\), then \((\mathcal{R}', S')\) are proofs of \(A \trianglelefteq B'\) with \(B \xrightarrow{q\text{Beffa}ra} B'\).

Proof.

Quasi-Beffaara preserves:

- being a proof-structure
- acyclicity of correctness graphs
- the number \(|V| + |\\gamma| - |E|\) of cc. of any correctness graph
- (result of composition over \(B\))
Completeness of Quasi-Beffara

Proposition

If $X \trianglelefteq B$ then $B \xrightarrow{\text{qBeffara}} \ast X$.

Proof.

By induction on the size of $B$. Trivial if $B = X$.
Else, by previous results:

1. we find some

2. which is a

3. $B \xrightarrow{\text{qBeffara}} B'$, $X \trianglelefteq B'$ and $B'$ of strictly smaller size
Remember Beffara’s retraction:

\[ X \triangleleft X \otimes (X^\perp \bowtie X) \quad X \triangleleft X \bowtie (X^\perp \otimes X) \]

Corresponding transformations inside a formula:

\[
\begin{align*}
X \otimes (X^\perp \bowtie X) & \xrightarrow{\text{Beffara}} X \\
X \bowtie (X^\perp \otimes X) & \xrightarrow{\text{Beffara}} X
\end{align*}
\]
Quasi-Beffara & Beffara (statement)

- Remember Beffara’s retraction:

\[ X \triangleleft X \otimes (X^\perp \triangleright X) \quad X \triangleleft X \triangleright (X^\perp \otimes X) \]

- Corresponding transformations inside a formula:

\[ X \otimes (X^\perp \triangleright X) \xrightarrow{\text{Beffara}} X \quad X \triangleright (X^\perp \otimes X) \xrightarrow{\text{Beffara}} X \]

**Proposition**

If \( q_{\text{Beffara}} \) \( X \), then \( B \xrightarrow{*} X \) up to isomorphism

(associativity and commutativity of \( \triangleright \) and \( \otimes \))
By induction on the size of $B$.

**Base cases:** $B \in \{X; X \rightarrow (X \perp \otimes X); X \otimes (X \perp \rightarrow X)\}$

**Inductive case:** $B \xrightarrow{\text{qBeffara}} B_1 \xrightarrow{\text{Beffara}} B_2 \xrightarrow{\text{Beffara}} * X$ by induction hypothesis.
Quasi-Beffara & Beffara (proof)

By induction on the size of $B$.

Base cases: $B \in \{X; X \otimes (X^\perp \otimes X); X \otimes (X^\perp \otimes X)\}$

Inductive case: $B \xrightarrow{\text{qBeffara}} B_1 \xrightarrow{\text{Beffara}} B_2 \xrightarrow{\text{Beffara}} \ast X$ by induction hypothesis.

$B \xrightarrow{\text{qBeffara}} B_1$ is

$B_1 \xrightarrow{\text{Beffara}} B_2$ is (up to duality)
Quasi-Beffara & Beffara (proof)

By induction on the size of $B$.

**Base cases:** $B \in \{X; X \not\in (X^\perp \otimes X); X \otimes (X^\perp \not\in X)\}$

**Inductive case:**

- $B \xrightarrow{\text{Beffara}} B_1 \xrightarrow{\text{Beffara}} B_2 \xrightarrow{\ast} X$ by induction hypothesis.

- $B \xrightarrow{\text{Beffara}} B_1$ is $X^\perp \otimes B$ or $X^\perp \otimes e_1$

- $B_1 \xrightarrow{\text{Beffara}} B_2$ is $X^\perp \otimes e_2$ (up to duality)

- $e_1 \notin \{a_1; a_2; a_3; a_4\}$ (including $e_1 = e_2$)

The rewritings commute: $B \xrightarrow{\text{Beffara}} B_1' \xrightarrow{\text{Beffara}} B_2 \xrightarrow{\ast} X$, so by induction $B \xrightarrow{\text{Beffara}} B_1' \xrightarrow{\ast} X$
Quasi-Beffara & Beffara (proof)

By induction on the size of $B$.

**Base cases:** $B \in \{X; X \otimes (X^\bot \otimes X); X \otimes (X^\bot \otimes X)\}$

**Inductive case:** $B \xrightarrow{\text{qBeffara}} B_1 \xrightarrow{\text{Beffara}} B_2 \xrightarrow{\text{Beffara}} \ast X$ by induction hypothesis.

$B \xrightarrow{\text{qBeffara}} B_1$ is or

$B_1 \xrightarrow{\text{Beffara}} B_2$ is (up to duality)

- $e_1 \not\in \{a_1; a_2; a_3; a_4\}$ (including $e_1 = e_2$) √
- $e_1 = a_2$
  - Up to isomorphism $e_1 = a_1$ or $e_1 = a_4$
Quasi-Beffara & Beffara (proof)

By induction on the size of $B$.

Base cases: $B \in \{X; X \not\otimes (X \perp \otimes X); X \otimes (X \perp \otimes X)\}$

Inductive case: $B$ by induction hypothesis.

$B \xrightarrow{q\text{Beffara}} B_1$ is $X \perp \otimes q\text{Beffara} \xrightarrow{\text{Beffara}} e_1$ or $X \otimes \perp e_1$

$B_1 \xrightarrow{\text{Beffara}} B_2$ is $X \otimes a_3 \otimes a_4 \xrightarrow{\text{Beffara}} e_2$ (up to duality)

- $e_1 \notin \{a_1; a_2; a_3; a_4\}$ (including $e_1 = e_2$)
- $e_1 = a_2$
- $e_1 \in \{a_1; a_3; a_4\}$

$B \xrightarrow{q\text{Beffara}} B_1$ is also a $B \xrightarrow{\text{Beffara}} B_1$
Characterization of $X \triangleleft B$

**Theorem**

The followings are equivalent:

1. $X \triangleleft B$
2. $B \xrightarrow{q_{\text{Beffa}}} ^* X$
3. $B \xrightarrow{\text{Beffa}} ^* X \ (\text{up to iso})$
Characterization of $X \triangleleft B$

**Theorem**

The followings are equivalent:

1. $X \triangleleft B$
2. $B \xrightarrow{q\text{Beffara}} * X$
3. $B \xrightarrow{\text{Beffara}} * X \ (\text{up to iso})$
4. $B \in P \ (\text{up to iso})$

\[ P ::= X \quad | \quad P \otimes (N \land P) \quad | \quad P \land (N \otimes P) \]

\[ N ::= X \perp \quad | \quad N \otimes (P \land N) \quad | \quad N \land (P \otimes N) \]
Characterization of $X \triangleleft B$

**Theorem**

The followings are equivalent:

1. $X \triangleleft B$
2. $B \xrightarrow{\text{qBeffara}} * X$
3. $B \xrightarrow{\text{Beffara}} * X$ (up to iso)
4. $B \in P$ (up to iso)

\[
P ::= X \mid P \otimes (N \otimes P) \mid P \otimes (N \otimes P) \\
N ::= X^\perp \mid N \otimes (P \otimes N) \mid N \otimes (P \otimes N)
\]

...but this is when looking at formulas! Looking at proofs, this is messier:

![Diagram showing various proof-structures and formulas](image-url)
Proof of $X \triangleleft (X \otimes X^\perp) \triangleright ((X \blacklozenge X^\perp) \otimes X^\perp)$

Not generated by Beffara as no $X^\perp \otimes X$ in either proof-nets
Incorrect retraction generated by Quasi-Beffara

Not-Proof of $X \triangleleft ((X \otimes (X \pitchfork X^\perp)) \pitchfork X^\perp) \otimes X$
Incorrect retraction generated by Quasi-Beffa

Not-Proof of $X \triangleleft ((X \otimes (X \otimes X)) \otimes X) \otimes X$

Incorrect
Incorrect retraction generated by Quasi-Beffara

Not-Proof of $X \triangleleft ((X \otimes (X \otimes X^\perp)) \otimes X^\perp) \otimes X$

Can apply one step of Quasi-Beffara
Incorrect retraction generated by Quasi-Beffara

Not-Proof of $X \triangleleft ((X \otimes (X \not\triangledown X)) \not\triangledown X) \otimes X$

This is Beffara, attainable from $X$ by one step of Quasi-Beffara
Formula not generated by Beffara without iso

\[ X \triangleleft X \otimes ((X \perp \ast X) \ast (X \perp \otimes X)) \]

Generated by Beffara only up to isomorphism!
Generalization to $A \triangleleft B$?

Not only $X \perp X$ as a pattern, also $X \perp \otimes X$; and maybe others?

Example: $X \otimes Y \leq X \otimes (X \perp \emptyset (X \otimes Y))$
Generalization to $A \triangleleft B$?

Not only as a pattern, also ; and maybe others?

Example: $X \otimes Y \triangleleft X \otimes (X \perp \triangleright (X \otimes Y))$
Plan

1. Multiplicative Linear Logic
   - Proof-Net
   - Retraction

2. Properties of Retractions

3. Difficulties for the general case $A \preceq B$

4. Retractions of the shape $X \sqsubseteq \cdot$ (universal super-types)
   - Looking for a pattern
   - Quasi-Beffara
   - Beffara $X \triangleleft X \otimes (X \bot \& X)$
   - Does not generalize to $A \preceq B$

5. Conclusion
What about the units? the $\text{mix}_2$-rule?

Result from [Balat and Di Cosmo, 1999]

Take $A$ and $B$ without sub-formulas of the shape $\neg \otimes 1$, $1 \otimes \neg$, $\bot \neg \neg$, nor $\neg \neg \bot$, and $\pi$ and $\pi'$ cut-free proofs respectively of $\vdash A \bot$, $B$ and $\vdash B \bot$, $A$. Then all $1$ and $\bot$-rules in $\pi$ and $\pi'$ belongs to the following pattern:

$$
\begin{array}{c}
\vdash 1 \\
\vdash \bot, 1 \\
\hline
\end{array}
$$

Thus $\left\{\begin{array}{l}
1 \rightarrow X \\
\bot \rightarrow X \bot
\end{array}\right.$ up to isomorphism

$\rightarrow$ same strict retractions with and without units; Cantor-Bernstein for MLL with units
What about the units? the $\text{mix}_2$-rule?

Result from [Balat and Di Cosmo, 1999]

Take $A$ and $B$ without sub-formulas of the shape $\neg \otimes 1$, $1 \otimes \neg$, $\bot \neg \neg$, nor $\neg \neg \bot$, and $\pi$ and $\pi'$ cut-free proofs respectively of $\vdash A \bot$, $B$ and $\vdash B \bot$, $A$. Then all $\top$ and $\bot$-rules in $\pi$ and $\pi'$ belongs to the following pattern:

$$
\begin{array}{c}
\top \bot \\
\top \top \\
\bot \bot
\end{array}
$$

Thus $\begin{cases} 
1 \rightarrow X \\
\bot \rightarrow X \bot
\end{cases}$ up to isomorphism

$\longrightarrow$ same strict retractions with and without units; Cantor-Bernstein for MLL with units

The $\text{mix}_2$-rule does not matter: it is preserved by composition and the identity has none.
Fact

\[ \begin{align*}
!X & \trianglelefteq !X \otimes !(X \otimes A) \iff \vdash A \text{ is provable} \\
X & \trianglelefteq X \& (X \otimes A) \iff \vdash A \text{ is provable} \\
A & \trianglelefteq A \oplus B \iff \vdash B^\bot, A \text{ is provable}
\end{align*} \]
Retractions and Provability

Fact

\[
\begin{align*}
!X &\trianglelefteqq !X \otimes !(X \otimes A) \iff \vdash A \text{ is provable} \\
X &\trianglelefteq X \& (X \otimes A) \iff \vdash A \text{ is provable} \\
A &\trianglelefteq A \oplus B \iff \vdash B \bot, A \text{ is provable}
\end{align*}
\]

<table>
<thead>
<tr>
<th>Fragment</th>
<th>Provability</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL</td>
<td>Undecidable 😞</td>
</tr>
<tr>
<td>MELL</td>
<td>TOWER-hard 😞 (decidability is open)</td>
</tr>
<tr>
<td>MALL</td>
<td>PSPACE-complete 😞</td>
</tr>
<tr>
<td>ALL</td>
<td>P-complete</td>
</tr>
</tbody>
</table>

(an overview of these results on provability can be found in [Lincoln, 1995])
\( X \sqsubseteq B \iff B \rightarrow^* X \) up to isomorphism with some subtleties on the proof-nets

- General properties: Cantor-Bernstein, result on sizes, only provability of a particular shape no consider, . . .
- Units, which are known for creating difficulties, do not matter here
- Still the problem is difficult!
  And it is even worse in larger fragments of linear logic.
\[ X \perp X \otimes (X \perp \mathcal{G} X) \quad A \perp A \otimes (X \perp \mathcal{G} X) \iff X \in A \]

\[ !X \perp !X \otimes !(X \otimes A) \iff \vdash A \quad X \perp X \& (X \otimes A) \iff \vdash A \]

Thank you for your attention!

\[ A \perp A \& B \iff \vdash A^\perp, B \quad A \perp A \oplus B \iff \vdash A, B^\perp \]

\[ X \oplus Y \perp ((X \oplus Z) \& (X \oplus Y)) \oplus Y \quad ?A \perp ??A \quad ?!A \perp ?!?A \]


Retractions of types with many atoms.

The category of finite sets and cartesian closed categories.
Back-up: What about other “simple” fragments?

- For exponential formulas, there are new retractions:
  
  \[ \text{?}A \subseteq \text{??}A \quad \text{?!}A \subseteq \text{?}!\text{?!}A \]

  Look like the only “basic” ones?
Back-up: What about other “simple” fragments?

- For **exponential** formulas, there are new retractions:

  \[ ?A \trianglelefteq ??A \quad ?!A \trianglelefteq ??A ?!?!A \]

  Look like the only “basic” ones?

- For **additive** formulas, only one “basic” retraction (with units too):

  \[ A \trianglelefteq A \& B \iff \vdash A \perp, B \quad \text{or} \quad A \trianglelefteq A \oplus B \iff \vdash A, B \perp \]

  Retraction of an atom manageable.

  But generally composition is bad due to the side condition:

  \[ X \oplus Y \trianglelefteq ((X \oplus Z) \& (X \oplus Y)) \oplus Y \]

  comes from \( X \oplus Y \trianglelefteq (X \oplus Y) \oplus Y \) without \( \vdash X \oplus Z, (X \oplus Y) \perp \)
Back-up: What about other “simple” fragments?

- For exponential formulas, there are new retractions:

  \(?A \trianglelefteq ??A\)  \(?!A \trianglelefteq ?!?!A\)

  Look like the only “basic” ones?

- For additive formulas, only one “basic” retraction (with units too):

  \(A \trianglelefteq A \land B \iff \vdash A\bot, B\) \quad \text{or} \quad \(A \trianglelefteq A \uplus B \iff \vdash A, B\bot\)

  Retraction of an atom manageable.

  But generally composition is bad due to the side condition:

  \(X \uplus Y \trianglelefteq ((X \uplus Z) \land (X \uplus Y)) \uplus Y\)

  comes from \(X \uplus Y \trianglelefteq (X \uplus Y) \uplus Y\) without \(\vdash X \uplus Z, (X \uplus Y)\bot\)

- Cantor-Bernstein holds in ALL. More complicated in MALL...