Retractions in Multiplicative Linear Logic

Rémi Di Guardia, Olivier Laurent

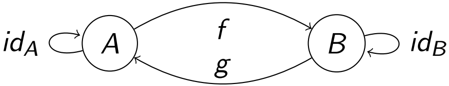
ENS Lyon (LIP)

Chocola 13/03/2024

Isomorphisms

Isomorphisms relate types/formulas/objects which are "the same"

$$A \simeq B$$



Instantiation in λ-calculus, logics,...

Wanted: an *equational theory*

Two main approaches:

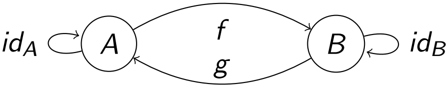
Syntactic the analysis of pairs of terms composing to the identity should provide information on their type

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

Isomorphisms relate types/formulas/objects which are "the same"

$$A \simeq B$$



For λ-calculus with products and unit type / cartesian closed categories

Semantic (finite sets) [Soloviev, 1983]

$\times$	$A \times (B \times C) \simeq (A \times B) \times C$	$A \times B \simeq B \times A$
$\times$ and $\to$	$(A \times B) \to C \simeq A \to (B \to C)$	$A \to (B \times C) \simeq (A \to B) \times (A \to C)$
1	$A \times 1 \simeq A \qquad 1 \to A \simeq A$	$A \to 1 \simeq 1$

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

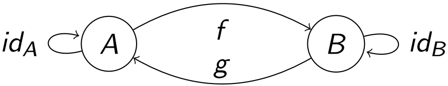
$$a(bc) = (ab)c \qquad ab = ba \qquad c^{ab} = (c^b)^a \qquad (bc)^a = b^a c^a$$

$$1a = a \qquad a^1 = a \qquad 1^a = 1$$

Isomorphisms

Isomorphisms relate types/formulas/objects which are "the same"

$$A \simeq B$$



For Multiplicative Linear Logic / $\star$-autonomous categories

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

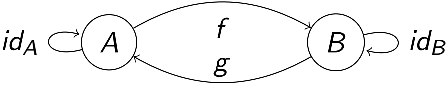
Associativity	$A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$	$A ⅋ (B ⅋ C) \simeq (A ⅋ B) ⅋ C$
Commutativity	$A \otimes B \simeq B \otimes A$	$A ⅋ B \simeq B ⅋ A$
Neutrality	$A \otimes 1 \simeq A$	$A ⅋ \bot \simeq A$

$$(A \otimes B) \multimap C = (A^\perp ⅋ B^\perp) ⅋ C \simeq A^\perp ⅋ (B^\perp ⅋ C) = A \multimap (B \multimap C)$$

Isomorphisms

Isomorphisms relate types/formulas/objects which are "the same"

$$A \simeq B$$



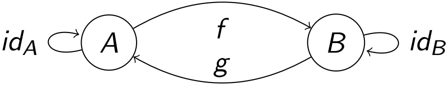
For Multiplicative-Additive Linear Logic / $\star$-autonomous categories with finite products   Syntactic (proof-nets) [Di Guardia and Laurent, 2023]

Associativity	$A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$ $A \oplus (B \oplus C) \simeq (A \oplus B) \oplus C$		$A \parr (B \parr C) \simeq (A \parr B) \parr C$ $A \& (B \& C) \simeq (A \& B) \& C$	
Commutativity	$A \otimes B \simeq B \otimes A$	$A \parr B \simeq B \parr A$	$A \oplus B \simeq B \oplus A$	$A \& B \simeq B \& A$
Neutrality	$A \otimes 1 \simeq A$	$A \parr \bot \simeq A$	$A \oplus 0 \simeq A$	$A \& \top \simeq A$
Distributivity	$A \otimes (B \oplus C) \simeq (A \otimes B) \oplus (A \otimes C)$		$A \parr (B \& C) \simeq (A \parr B) \& (A \parr C)$	
Annihilation	$A \otimes 0 \simeq 0$		$A \parr \top \simeq \top$	

Isomorphisms

Isomorphisms relate types/formulas/objects which are "the same"

$$A \simeq B$$



For Polarized Linear Logic

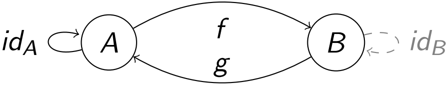
Semantic (games, forest isomorphisms) [Laurent, 2005]

Associativity	$A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$ $A \oplus (B \oplus C) \simeq (A \oplus B) \oplus C$		$A \parr (B \parr C) \simeq (A \parr B) \parr C$ $A \,\&\, (B \,\&\, C) \simeq (A \,\&\, B) \,\&\, C$	
Commutativity	$A \otimes B \simeq B \otimes A$	$A \parr B \simeq B \parr A$	$A \oplus B \simeq B \oplus A$	$A \,\&\, B \simeq B \,\&\, A$
Neutrality	$A \otimes 1 \simeq A$	$A \parr \bot \simeq A$	$A \oplus 0 \simeq A$	$A \,\&\, \top \simeq A$
Distributivity	$A \otimes (B \oplus C) \simeq (A \otimes B) \oplus (A \otimes C)$		$A \parr (B \,\&\, C) \simeq (A \parr B) \,\&\, (A \parr C)$	
Annihilation	$A \otimes 0 \simeq 0$		$A \parr \top \simeq \top$	
Seely	$!(A \,\&\, B) \simeq !A \otimes !B$ $!\top \simeq 1$		$?(A \oplus B) \simeq ?A \parr ?B$ $?0 \simeq \bot$	

Retractions

Retractions relate A and B when A is a "sub-type" of B

$$A \trianglelefteq B$$



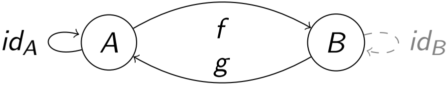
Instantiation in λ-calculus, logics,...

bool $\trianglelefteq$ nat with $f(\text{false}) = 0$, $f(\text{true}) = 1$ and $g(n) = n$ is equal to 1

Retractions

Retractions relate A and B when A is a "sub-type" of B

$$A \trianglelefteq B$$



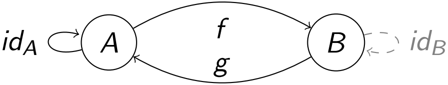
Instantiation in λ-calculus, logics,...

bool $\trianglelefteq$ nat with $f(\text{false}) = 0$, $f(\text{true}) = 42$ and $g(n) = n$ is equal to 42

Retractions

Retractions relate A and B when A is a "sub-type" of B

$$A \trianglelefteq B$$



For simply typed affine λ-calculus

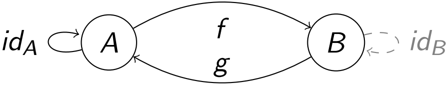
Syntactic [Regnier and Urzyczyn, 2002]

$\simeq$	$A \to B \to C \simeq B \to A \to C$
$\lhd \ \ (= \trianglelefteq \setminus \simeq)$	$A \lhd B \to A$ $A \lhd (A \to X) \to X$ if A is $Y_1 \to Y_2 \to \cdots \to X$

Retractions

Retractions relate A and B when A is a "sub-type" of B

$$A \trianglelefteq B$$



For Multiplicative Linear Logic

[UNKNOWN]

$\simeq$	associativity and commutativity of $\otimes$ and $⅋$, neutrality of 1 and $\perp$
$\triangleleft \quad (= \trianglelefteq \setminus \simeq)$	???

Other results about retractions

Decidability of retractions in simply typed λ-calculus in [Padovani, 2001]

Definition

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

Holds in some category but not all!

Plan

1. Definitions
   - 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 $X \lhd X \otimes (X^\perp \parr X)$
   - Does not generalize to $A \trianglelefteq B$
5. Conclusion

Formula & Sequent

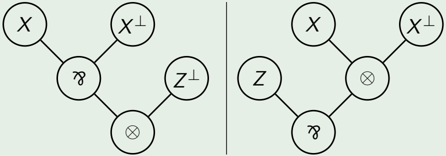
Formula

$$A, B \;::=\; X \mid \overset{not}{X^\perp} \mid A \overset{and}{\otimes} B \mid A \overset{or}{⅋} B$$

Duality

$$(X^\perp)^\perp = X$$
$$(A \otimes B)^\perp = B^\perp ⅋ A^\perp$$
$$(A ⅋ B)^\perp = B^\perp \otimes A^\perp$$

Examples



Formula & Sequent

Formula

$$A, B \;::=\; X \mid \overset{not}{X^\perp} \mid A \overset{and}{\otimes} B \mid A \overset{or}{⅋} B$$

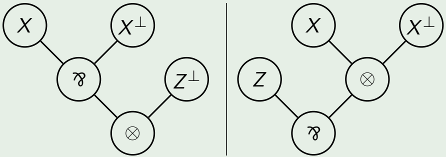
Duality

$$(X^\perp)^\perp = X$$
$$(A \otimes B)^\perp = B^\perp ⅋ A^\perp$$
$$(A ⅋ B)^\perp = B^\perp \otimes A^\perp$$

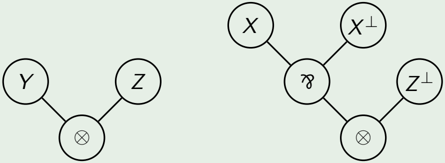
Sequent

$$\vdash A_1, \ldots, A_n$$

Examples



Example



Formula & Sequent

Formula

$$A, B \;::=\; X \mid \overset{not}{X^\perp} \mid A \overset{and}{\otimes} B \mid A \overset{or}{\parr} B$$

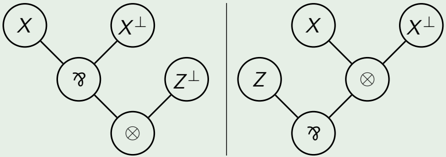
Duality

$$(X^\perp)^\perp = X$$
$$(A \otimes B)^\perp = B^\perp \parr A^\perp$$
$$(A \parr B)^\perp = B^\perp \otimes A^\perp$$

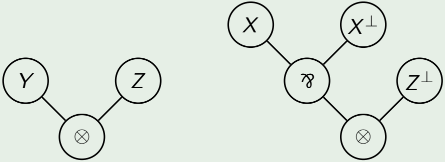
Sequent

$$\vdash A_1, \ldots, A_n$$

Examples



Example



Rules (sequent calculus)

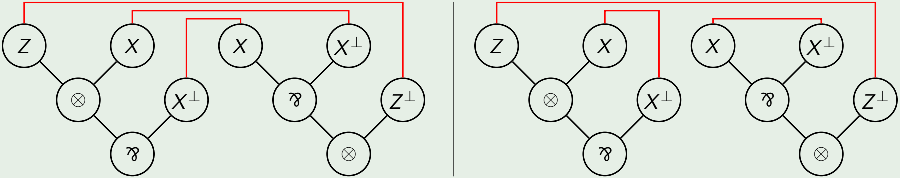
$$\frac{}{\vdash A^\perp, A}\,ax \qquad \frac{\vdash A, \Gamma \quad \vdash B, \Delta}{\vdash A \otimes B, \Gamma, \Delta}\,\otimes \qquad \frac{\vdash A, B, \Gamma}{\vdash A \parr B, \Gamma}\,\parr$$

Proof-Structure

Proof-Structure

Sequent $\vdash A, B$ with edges between dual leaves (some X and $X^\perp$), these edges partitioning the leaves of the sequent.

Examples

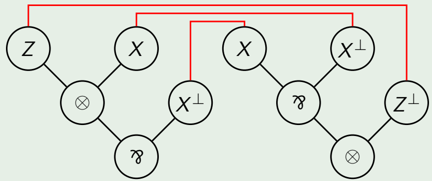


Proof-Structure

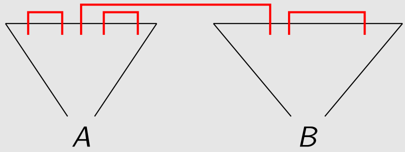
Proof-Structure

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

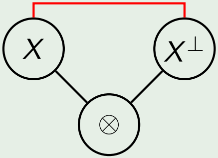
Correctness Graph

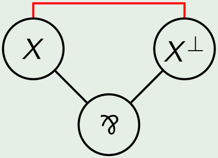
In a proof-structure, keep only one premise of each ⅋-node.

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





Correctness & Proof-Net

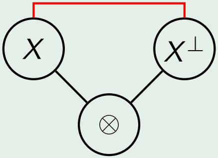
Correctness Graph

In a proof-structure, keep only one premise of each ⅋-node.

Danos-Regnier Correctness Criterion

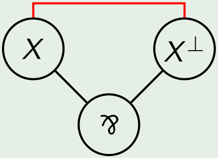
A proof-structure is *correct*, and called a *proof-net*, if all its correctness graphs are acyclic and connected (i.e. are trees).

Toy examples



Not acyclic (but connected)

INCORRECT



Correctness & Proof-Net

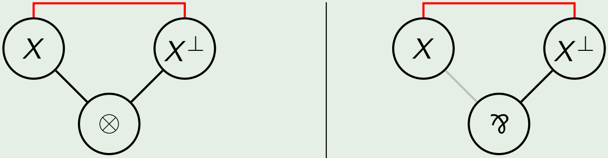
Correctness Graph

In a proof-structure, keep only one premise of each ⅋-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

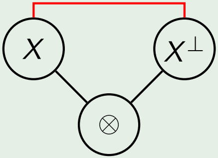
Correctness Graph

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

Danos-Regnier Correctness Criterion

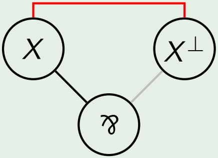
A proof-structure is *correct*, and called a *proof-net*, if all its correctness graphs are acyclic and connected (i.e. are trees).

Toy examples



Not acyclic (but connected)

INCORRECT



Acyclic and connected

CORRECT

Correctness & Proof-Net

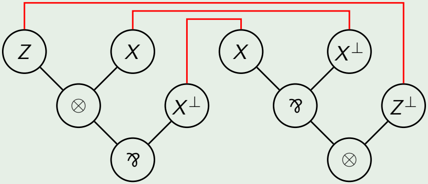
Correctness Graph

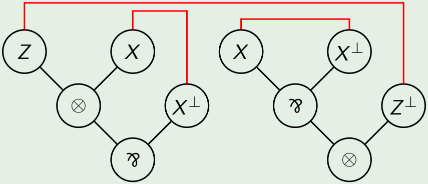
In a proof-structure, keep only one premise of each ⅋-node.

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





Correctness & Proof-Net

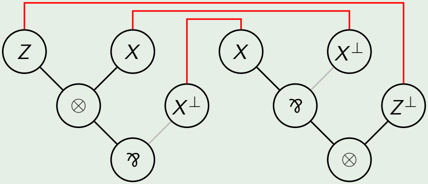
Correctness Graph

In a proof-structure, keep only one premise of each ⅋-node.

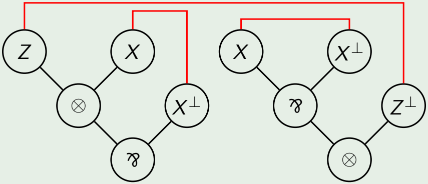
Danos-Regnier Correctness Criterion

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 & Proof-Net

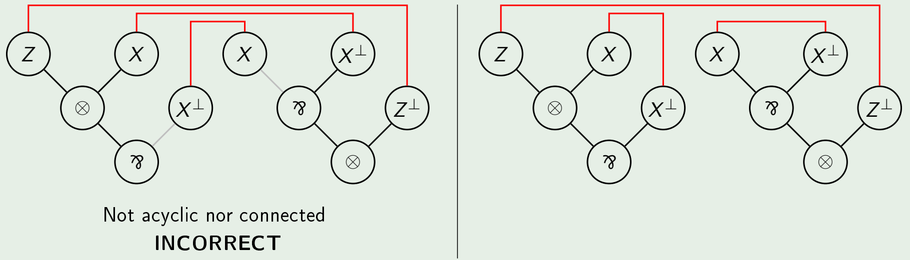
Correctness Graph

In a proof-structure, keep only one premise of each ⅋-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

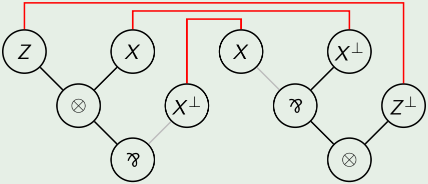
Correctness Graph

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

Danos-Regnier Correctness Criterion

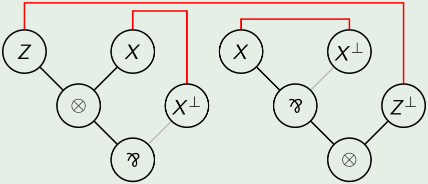
A proof-structure is *correct*, and called a *proof-net*, if all its correctness graphs are acyclic and connected (i.e. are trees).

Examples



Not acyclic nor connected

INCORRECT



Acyclic and connected

Correctness & Proof-Net

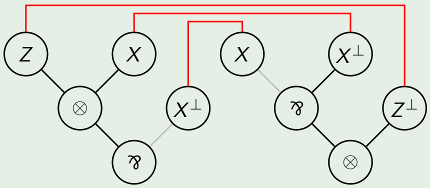
Correctness Graph

In a proof-structure, keep only one premise of each ⅋-node.

Danos-Regnier Correctness Criterion

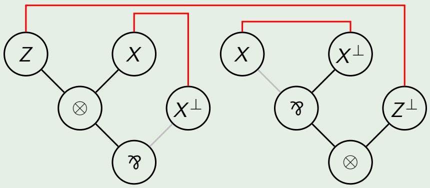
A proof-structure is *correct*, and called a *proof-net*, if all its correctness graphs are acyclic and connected (i.e. are trees).

Examples



Not acyclic nor connected

INCORRECT



Acyclic and connected

Correctness & Proof-Net

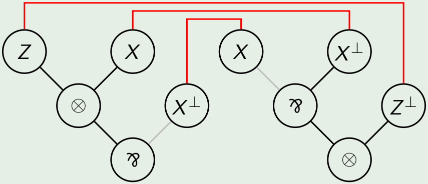
Correctness Graph

In a proof-structure, keep only one premise of each ⅋-node.

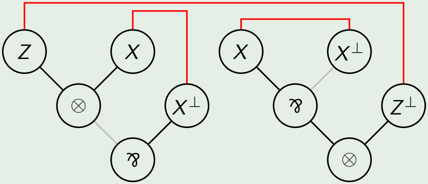
Danos-Regnier Correctness Criterion

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

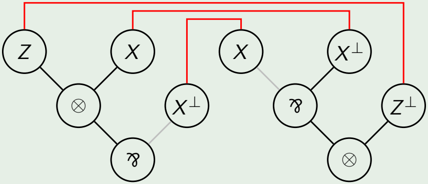
Correctness Graph

In a proof-structure, keep only one premise of each ⅋-node.

Danos-Regnier Correctness Criterion

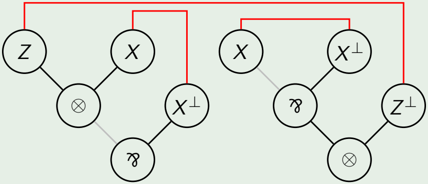
A proof-structure is *correct*, and called a *proof-net*, if all its correctness graphs are acyclic and connected (i.e. are trees).

Examples



Not acyclic nor connected

INCORRECT



Acyclic and connected

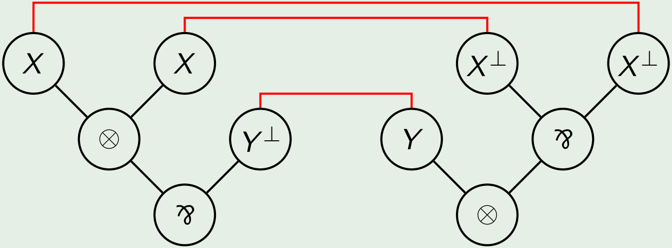
CORRECT

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



Lemma

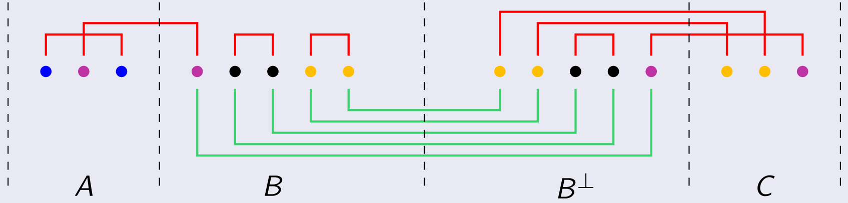
An identity proof-structure is correct.

Composition

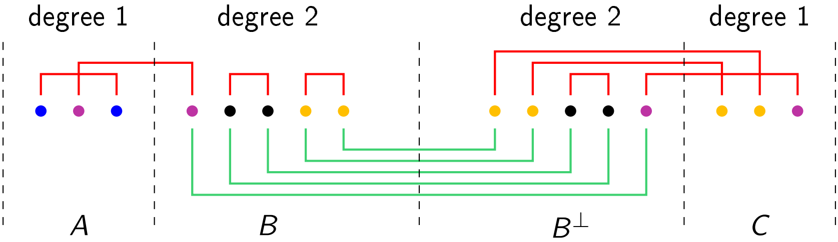
Equivalence Class of a leaf

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.



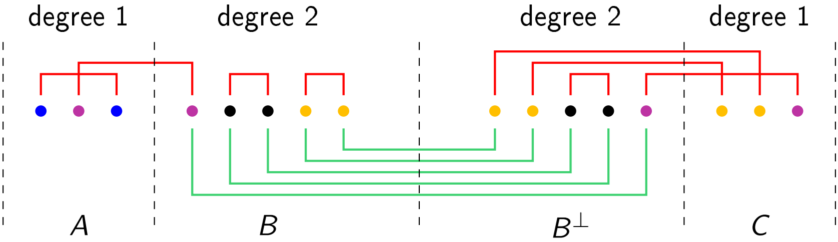
Composition bis



Lemma

A graph containing only vertices of degree 1 or 2 is a disjoint union of non-empty simple paths and cycles.

Composition bis



Lemma

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. *(But we do not need it here.)*

Composition ter

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.

Composition ter

Composition

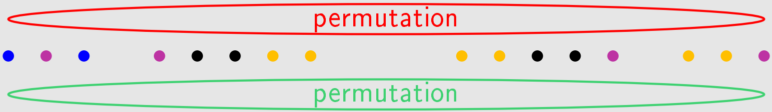
Take two proof-structures on $\vdash A, B$ and $\vdash B^\perp, C$. Delete edges involving leaves of B and $B^\perp$ and add edges between leaves of A and B in the same equivalence class, obtaining a proof-structure on $\vdash A, C$.

Lemma

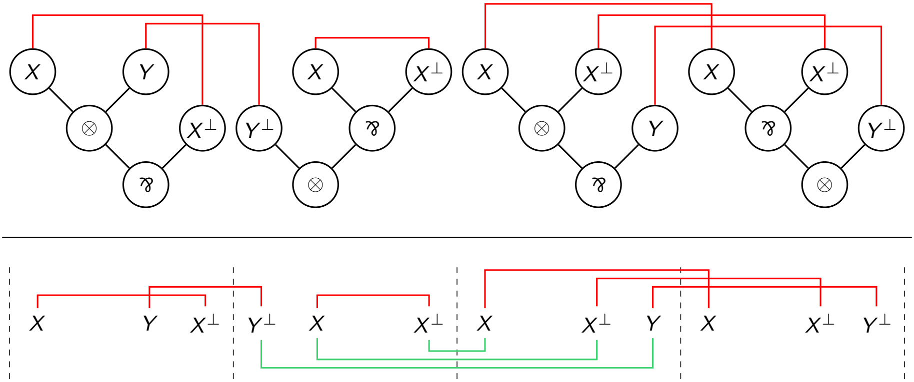
The composition of two proof-nets is a proof-net.

Orthogonality of GOI / of Danos-Regnier

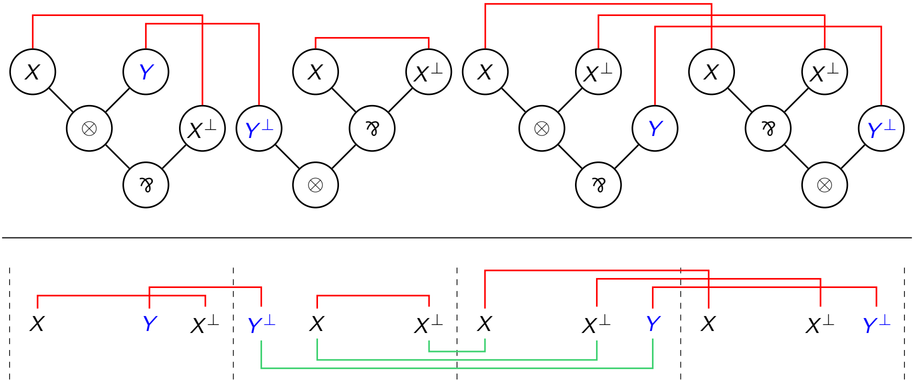
Composition of permutations, yielding a permutation if they are orthogonal = there are no cycles, only paths



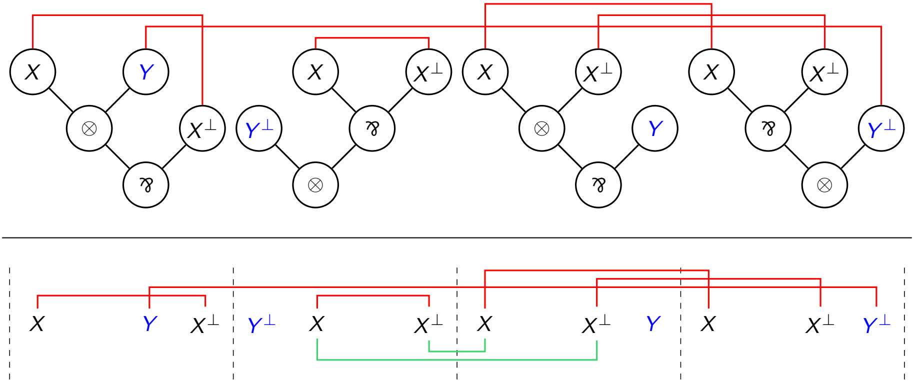
Example of composition



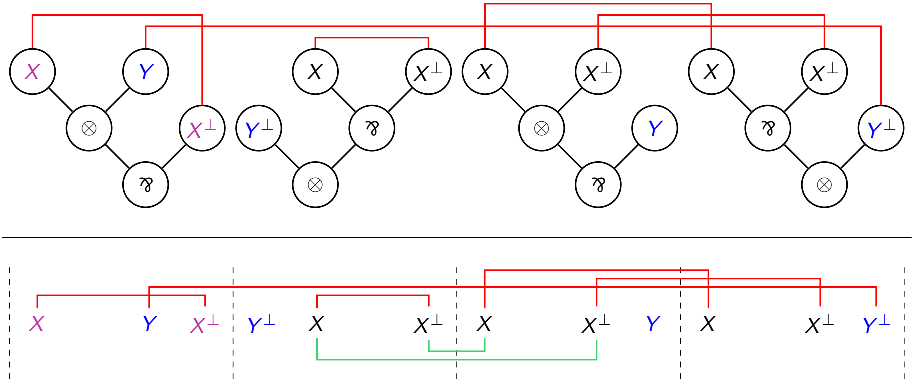
Example of composition



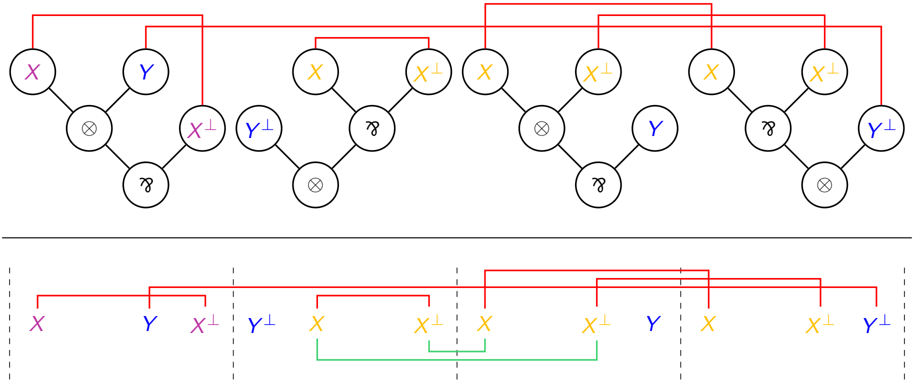
Example of composition



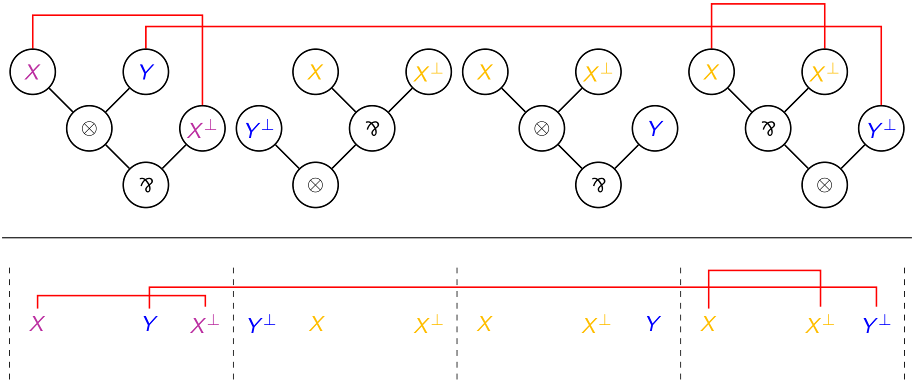
Example of composition



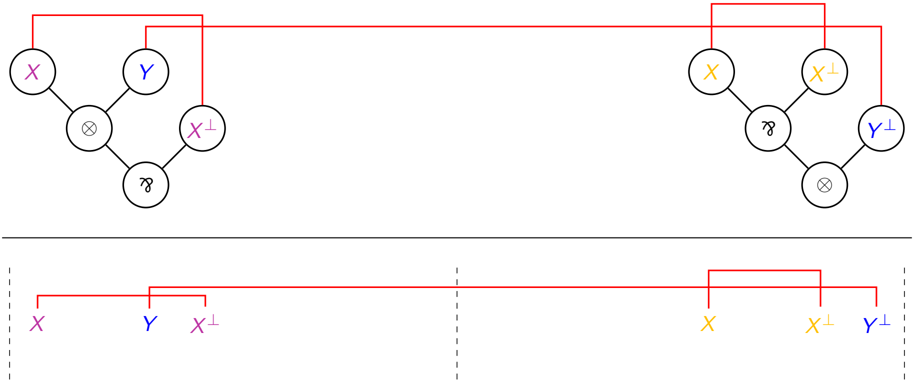
Example of composition



Example of composition

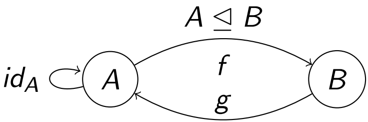


Example of composition



Retraction

Category theory



λ-calculus

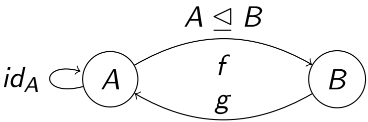
Retraction $A \trianglelefteq B$

Terms $M : A \to B$ and $N : B \to A$ such that

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

Retraction

Category theory



λ-calculus

Retraction $A \trianglelefteq B$

Terms $M : A \to B$ and $N : B \to A$ such that

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

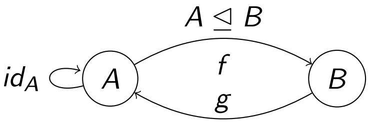
Multiplicative Linear Logic

Retraction $A \trianglelefteq B$

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



λ-calculus

Retraction $A \trianglelefteq B$

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 $A \trianglelefteq B$

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.

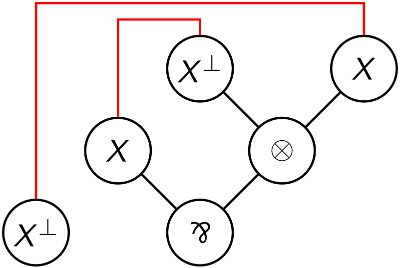
$$A \trianglelefteq B \iff A^\perp \trianglelefteq B^\perp$$

Beffara's retraction

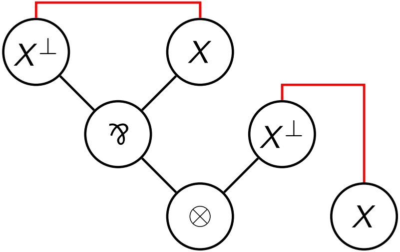
Beffara's retraction

$$X \lhd X \parr (X^\perp \otimes X) \qquad \text{or dualy} \qquad X \lhd X \otimes (X^\perp \parr X)$$

Can also be seen as $X \lhd (X \multimap X) \multimap X$

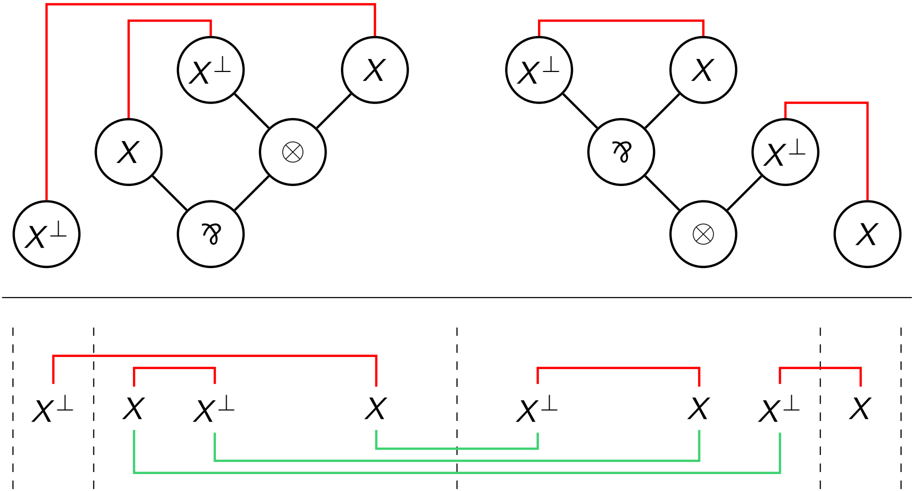


$X \parr (X^\perp \otimes X)$

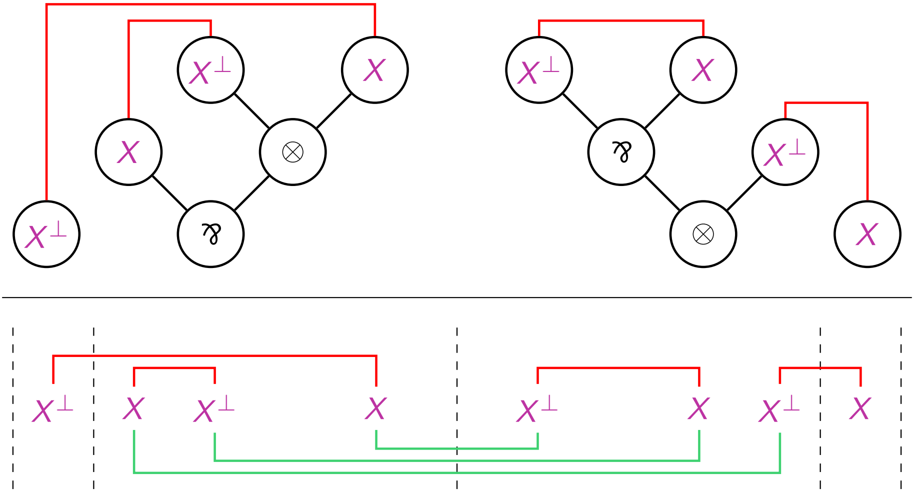


$(X^\perp \parr X) \otimes X^\perp$

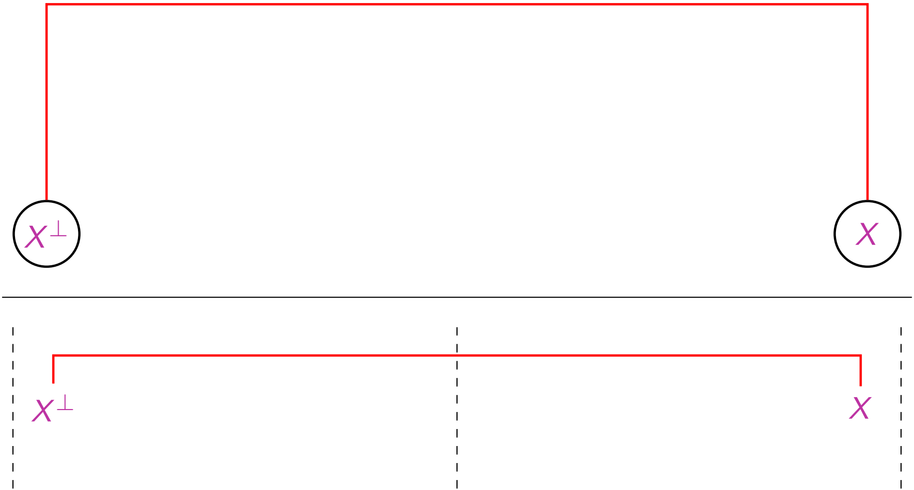
Beffara's is a retraction



Beffara's is a retraction



Beffara's is a retraction



Plan

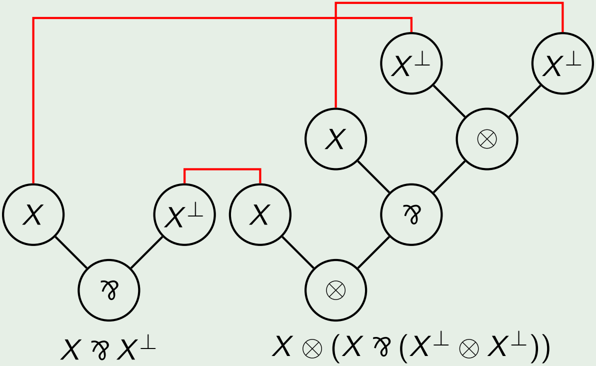
Half-Bipartiteness

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 $X \parr X^\perp$ but not in $X \otimes (X \parr (X^\perp \otimes X^\perp))$.



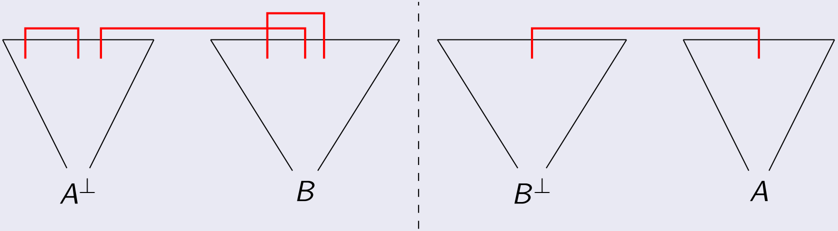
Retractions are half-bipartite

Lemma

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

Proof.

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



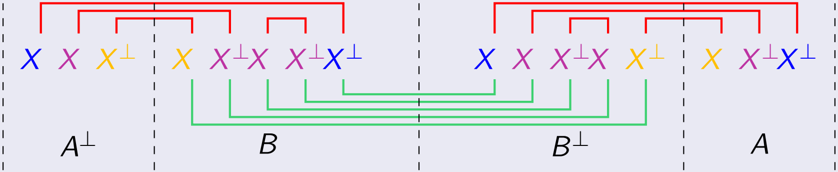
$\square$

Non-ambiguity

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.



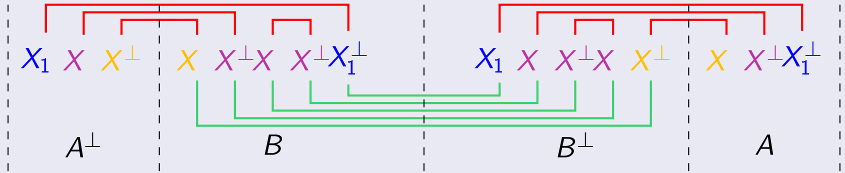
Rename each equivalence class with a fresh atom.

Non-ambiguity

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.



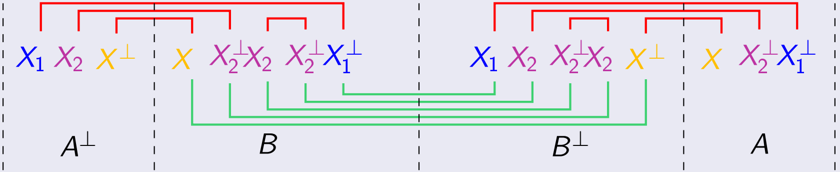
Rename each equivalence class with a fresh atom.

Non-ambiguity

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.



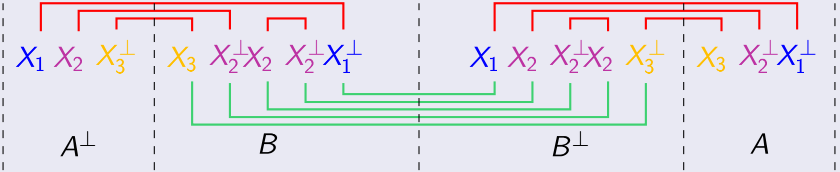
Rename each equivalence class with a fresh atom.

Non-ambiguity

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.



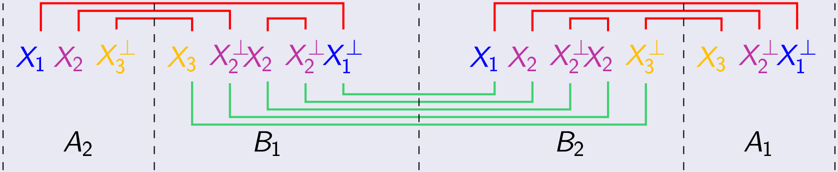
Rename each equivalence class with a fresh atom.

Non-ambiguity

Corollary: Non-ambiguity

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

Proof.



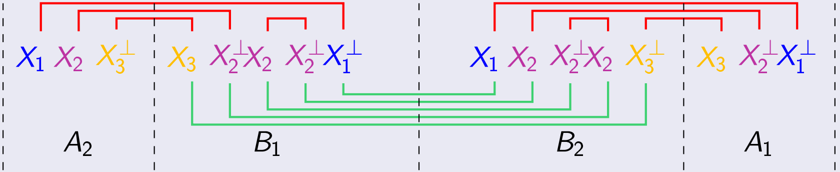
Rename each equivalence class with a fresh atom.

Non-ambiguity

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.

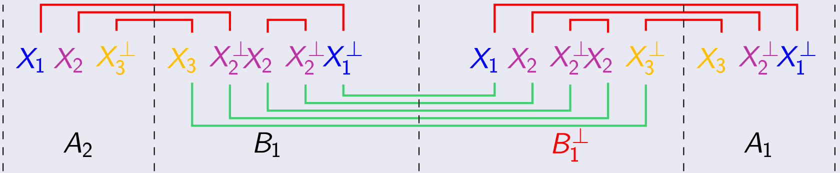
1. Half-bipartiteness $\to$ one atom of A by class, so A_1 non-ambiguous

Non-ambiguity

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.

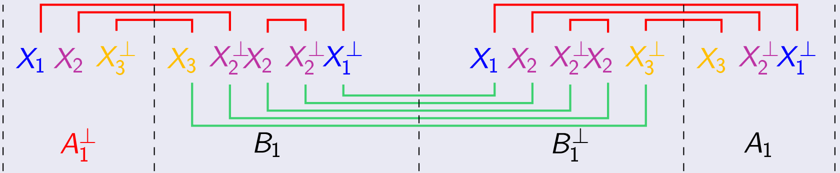
1. Half-bipartiteness → one atom of A by class, so A_1 non-ambiguous
2. Dual leaves of B and $B^\perp$ in the same equivalence class → $B_2 = B_1^\perp$

Non-ambiguity

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.

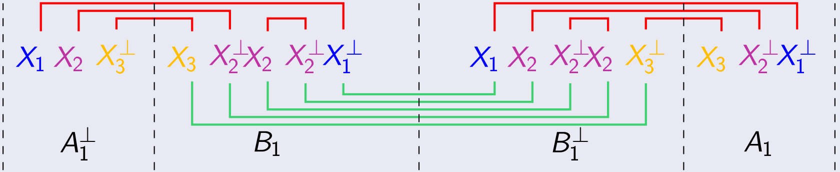
1. Half-bipartiteness → one atom of A by class, so A_1 non-ambiguous
2. Dual leaves of B and $B^\perp$ in the same equivalence class → $B_2 = B_1^\perp$
3. Composition is identity → dual leaves of $A^\perp$ and A in the same equivalence class → $A_2 = A_1^\perp$

Non-ambiguity

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. Half-bipartiteness → one atom of A by class, so A_1 non-ambiguous
2. Dual leaves of B and $B^\perp$ in the same equivalence class → $B_2 = B_1^\perp$
3. Composition is identity → dual leaves of $A^\perp$ and A in the same equivalence class → $A_2 = A_1^\perp$
4. 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 $A \trianglelefteq B$ 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~~.

Property on sizes

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

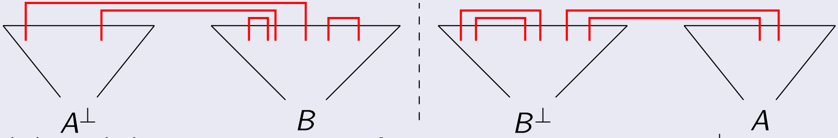
Retraction $A \trianglelefteq B$ with A non-ambiguous

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 $A \trianglelefteq B$, 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]:

Associativity	$A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$	$A \parr (B \parr C) \simeq (A \parr B) \parr C$
Commutativity	$A \otimes B \simeq B \otimes A$	$A \parr B \simeq B \parr A$

Consequences

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

Associativity	$A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$	$A \parr (B \parr C) \simeq (A \parr B) \parr C$
Commutativity	$A \otimes B \simeq B \otimes A$	$A \parr B \simeq B \parr A$

Corollary

The Cantor-Bernstein property holds:

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

Consequences

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

Associativity	$A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$	$A \parr (B \parr C) \simeq (A \parr B) \parr C$
Commutativity	$A \otimes B \simeq B \otimes A$	$A \parr B \simeq B \parr A$

Corollary

The Cantor-Bernstein property holds:

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

$$X \otimes Y \ntrianglelefteq X \parr Y$$

$$X \parr (Y \otimes Z) \ntrianglelefteq Y \otimes (X \parr Z)$$

Plan

Not finitely axiomatisable?

$$X_1 \otimes X_2 \otimes X_3 \otimes X_4 \lhd (X_1 \otimes X_2 \otimes X_3 \otimes X_4) \parr (X_1 \otimes (X_1^\perp \parr (X_2 \otimes (X_2^\perp \parr (X_3 \otimes (X_3^\perp \parr (X_4 \otimes X_4^\perp)))))))$$

Generally:

$$\{\otimes X_i\} \lhd \{\otimes X_i\} \parr (X_1 \otimes (X_1^\perp \parr (\dots (X_{n-1} \otimes (X_{n-1}^\perp \parr (X_n \otimes X_n^\perp)) \dots)))$$

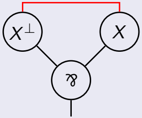
However $(A \otimes X) \parr B \ntrianglelefteq (A \otimes X) \parr (X \otimes (X^\perp \parr B))$

Plan

Key Result

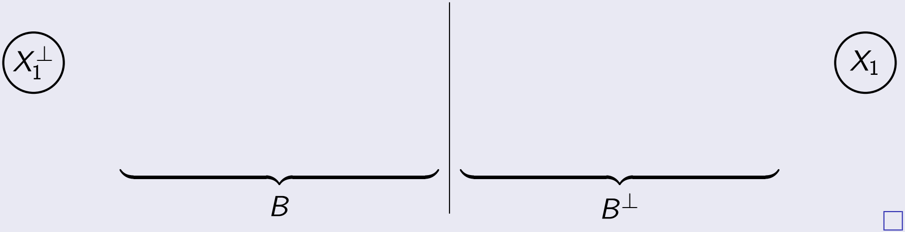
Lemma

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



Proof.

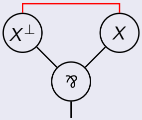
We build a sequence (GOI path) finding such a pattern.



□

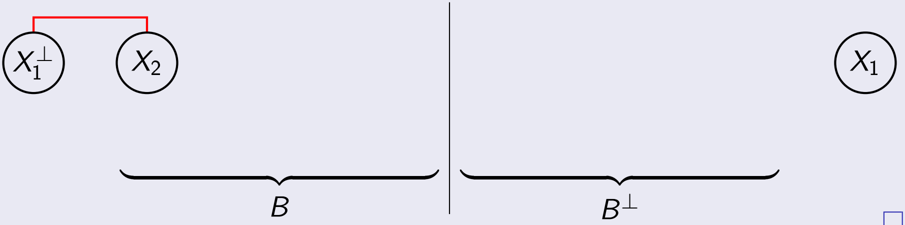
Lemma

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



Proof.

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

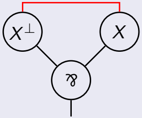


□

Key Result

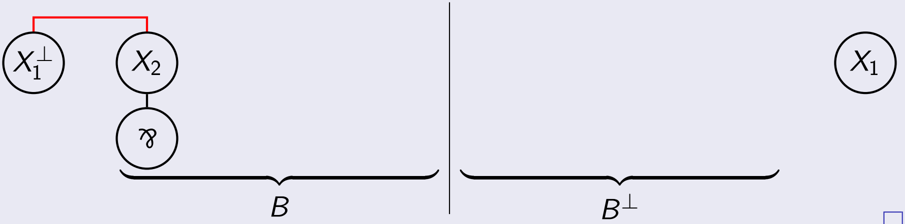
Lemma

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



Proof.

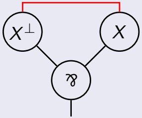
We build a sequence (GOI path) finding such a pattern.



□

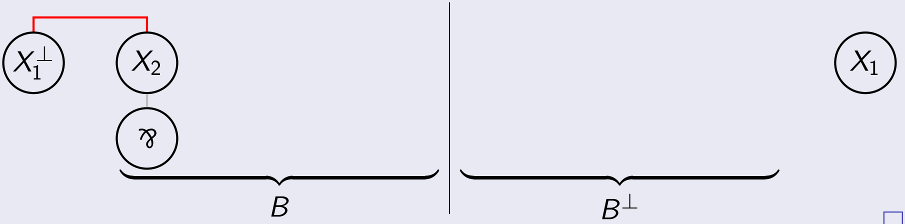
Key Result

Lemma

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

Proof.

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

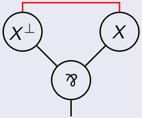


□

Key Result

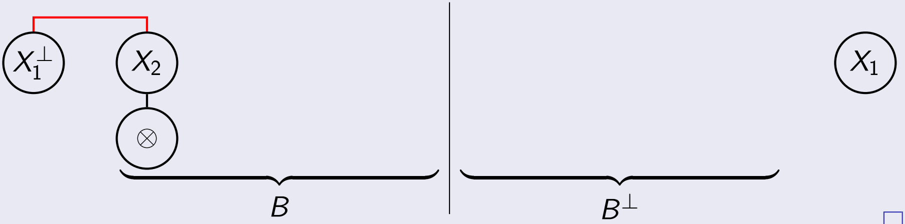
Lemma

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



Proof.

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

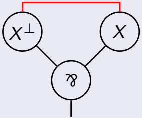


□

Key Result

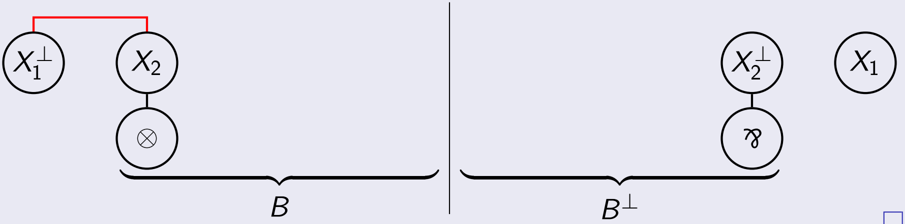
Lemma

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



Proof.

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

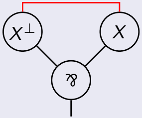


□

Key Result

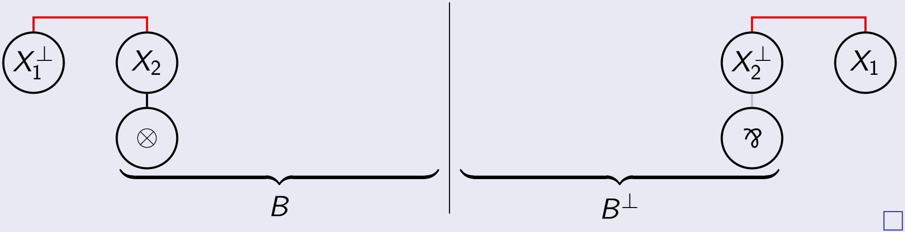
Lemma

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



Proof.

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

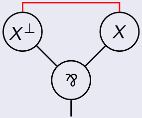


□

Key Result

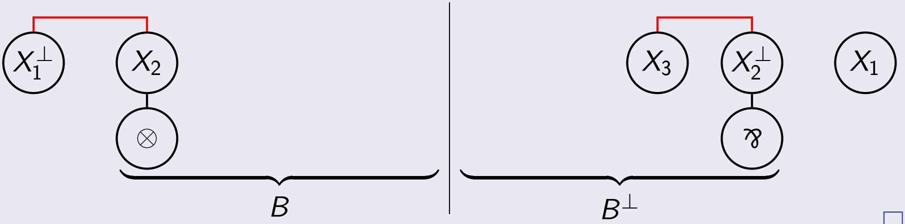
Lemma

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



Proof.

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

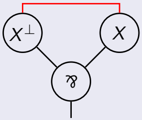


□

Key Result

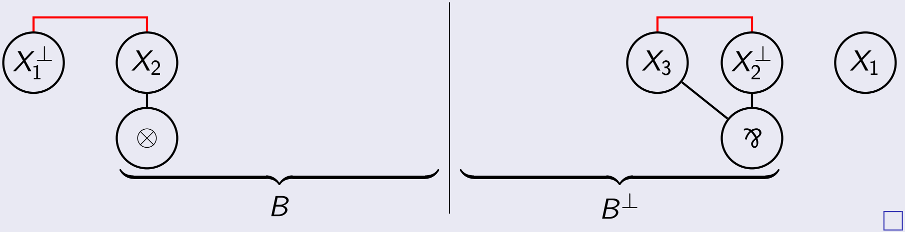
Lemma

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



Proof.

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

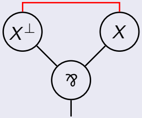


□

Key Result

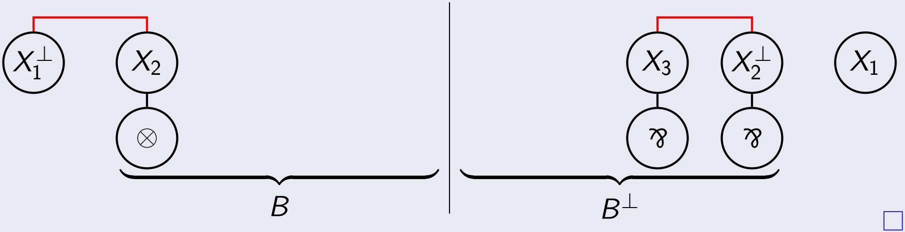
Lemma

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



Proof.

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

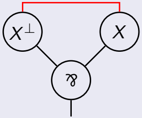


□

Key Result

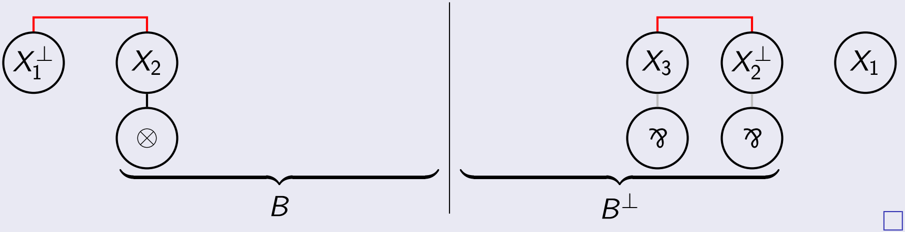
Lemma

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



Proof.

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

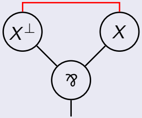


□

Key Result

Lemma

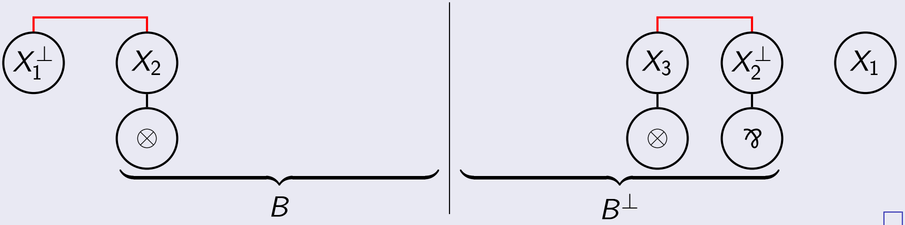
In $X \lhd B$ one of the two proof-nets contains:



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 $\parr$.

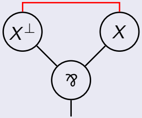


□

Key Result

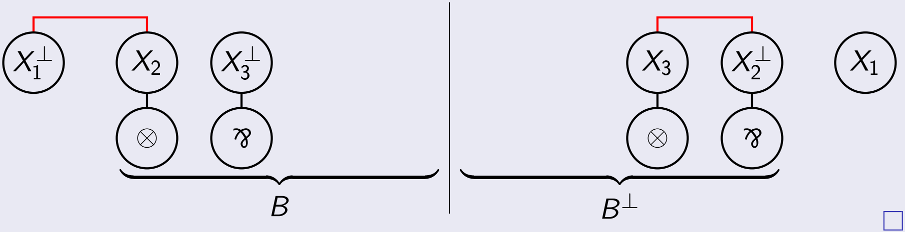
Lemma

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



Proof.

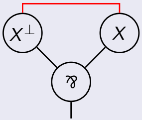
We build a sequence (GOI path) finding such a pattern.
Invariant: every X of B is above a $\otimes$, and every $X^\perp$ above a ⅋.



Key Result

Lemma

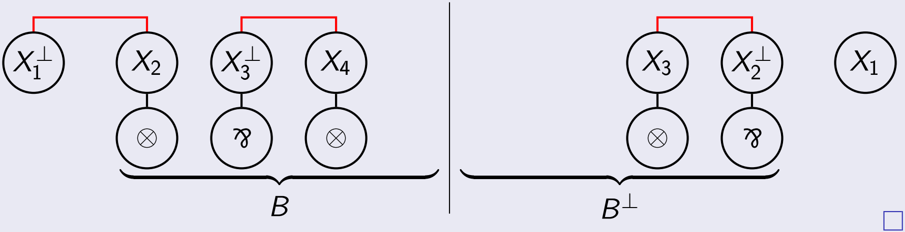
In $X \lhd B$ one of the two proof-nets contains:



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 ⅋.

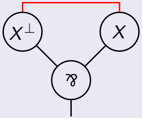


□

Key Result

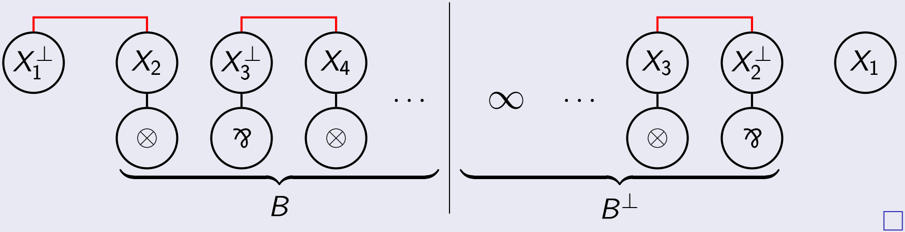
Lemma

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



Proof.

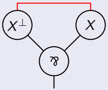
We build a sequence (GOI path) finding such a pattern.
Invariant: every X of B is above a $\otimes$, and every $X^\perp$ above a $\parr$.

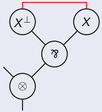


□

Extended pattern

Lemma

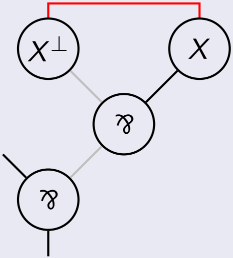
If 

has a node below it, then this is a 

.

Proof.

The connector below the pattern cannot be a ⅋ by connectivity:

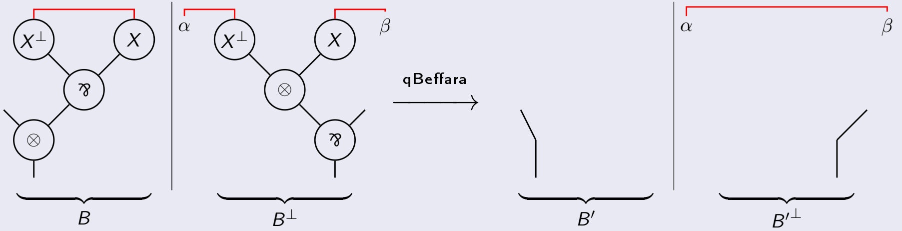


□

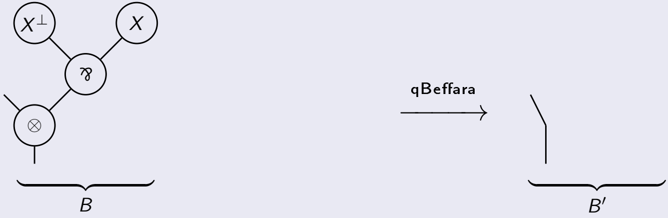
Quasi-Beffara

Definition

Quasi-Beffara is this local transformation on proofs of a retraction $A \trianglelefteq B$:



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



Coherence of Quasi-Beffara

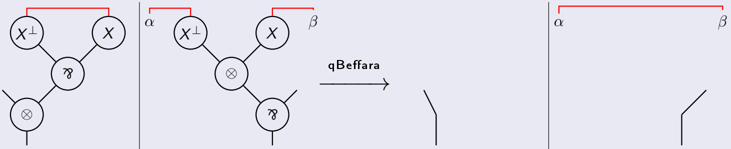
Lemma

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

Proof.

Quasi-Beffara preserves:

- being a proof-structure



□

Coherence of Quasi-Beffara

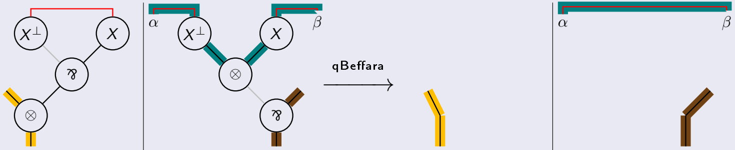
Lemma

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

Proof.

Quasi-Beffara preserves:

- being a proof-structure
- acyclicity of correctness graphs



□

Coherence of Quasi-Beffara

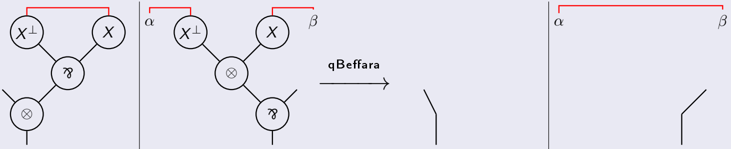
Lemma

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

Proof.

Quasi-Beffara preserves:

- being a proof-structure
- acyclicity of correctness graphs
- the number $|V| + |\parr| - |E|$ of cc. of any correctness graph: it removes 4 vertices, including 1 $\parr$, and 5 edges



□

Coherence of Quasi-Beffara

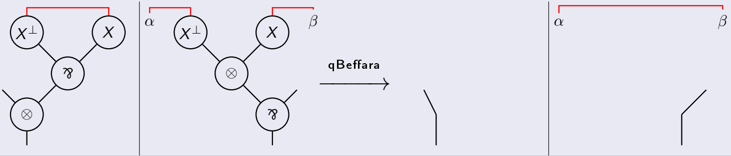
Lemma

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

Proof.

Quasi-Beffara preserves:

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



∎

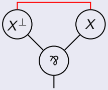
Completeness of Quasi-Beffara

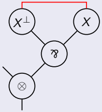
Proposition

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

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

Quasi-Beffara & Beffara (statement)

- Remember Beffara's retraction:

$$X \lhd X \otimes (X^\perp \parr X) \qquad\qquad X \lhd X \parr (X^\perp \otimes X)$$

- Corresponding transformations inside a formula:

$$X \otimes (X^\perp \parr X) \xrightarrow{\text{Beffara}} X \qquad\qquad X \parr (X^\perp \otimes X) \xrightarrow{\text{Beffara}} X$$

Quasi-Beffara & Beffara (statement)

- Remember Beffara's retraction:

$$X \lhd X \otimes (X^\perp \parr X) \qquad X \lhd X \parr (X^\perp \otimes X)$$

- Corresponding transformations inside a formula:

$$X \otimes (X^\perp \parr X) \xrightarrow{\text{Beffara}} X \qquad X \parr (X^\perp \otimes X) \xrightarrow{\text{Beffara}} X$$

Proposition

*If $B \xrightarrow{\text{qBeffara}}{}^* X$, then $B \xrightarrow{\text{Beffara}}{}^* X$ **up to isomorphism***
(associativity and commutativity of $\parr$ and $\otimes$)

Quasi-Beffara & Beffara (proof)

By induction on the size of B.

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

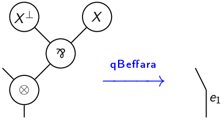
Inductive case: $B \xrightarrow{\text{qBeffara}} B_1 \xrightarrow{\text{Beffara}} B_2 \xrightarrow{\text{Beffara}}{}^{*} X$ by induction hypothesis.

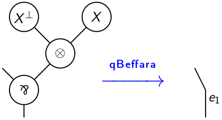
Quasi-Beffara & Beffara (proof)

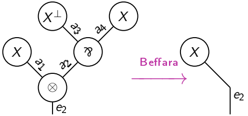
By induction on the size of B.

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

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

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

or 

$B_1 \xrightarrow{\text{Beffara}} B_2$ is 

(up to duality)

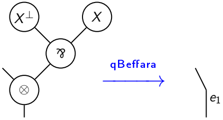
Quasi-Beffara & Beffara (proof)

By induction on the size of B.

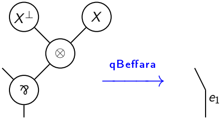
<u>Base cases:</u> $B \in \{X; X \parr (X^\perp \otimes X); X \otimes (X^\perp \parr X)\}$

<u>Inductive case:</u> $B \xrightarrow{\text{qBeffara}} B_1 \xrightarrow{\text{Beffara}} B_2 \xrightarrow{\text{Beffara}}{}^{*} X$ by induction hypothesis.

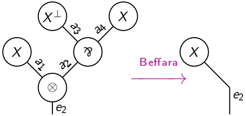
$B \xrightarrow{\text{qBeffara}} B_1$ is



or



$B_1 \xrightarrow{\text{Beffara}} B_2$ is



(up to duality)

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

  The rewritings commute: $B \xrightarrow{\text{Beffara}} B_1' \xrightarrow{\text{qBeffara}} B_2 \xrightarrow{\text{Beffara}}{}^{*} X$, so by induction $B \xrightarrow{\text{Beffara}} B_1' \xrightarrow{\text{Beffara}}{}^{*} X$

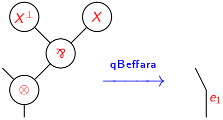
Quasi-Beffara & Beffara (proof)

By induction on the size of B.

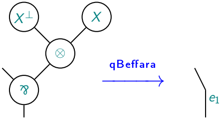
<u>Base cases:</u> $B \in \{X ; X \parr (X^\perp \otimes X) ; X \otimes (X^\perp \parr X)\}$

<u>Inductive case:</u> $B \xrightarrow{\text{qBeffara}} B_1 \xrightarrow{\text{Beffara}} B_2 \xrightarrow{\text{Beffara}}{}^{*} X$ by induction hypothesis.

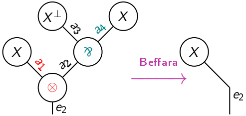
$B \xrightarrow{\text{qBeffara}} B_1$ is



or



$B_1 \xrightarrow{\text{Beffara}} B_2$ is



(up to duality)

- $e_1 \notin \{a_1 ; a_2 ; a_3 ; a_4\}$ (including $e_1 = e_2$) $\surd$
- $e_1 = a_2$
  Up to isomorphism $e_1 = a_1$ or $e_1 = a_4$

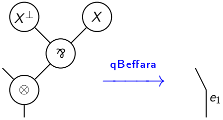
Quasi-Beffara & Beffara (proof)

By induction on the size of B.

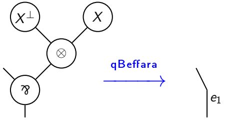
Base cases: $B \in \{X; X \parr (X^\perp \otimes X); X \otimes (X^\perp \parr X)\}$

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

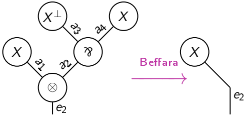
$B \xrightarrow{\text{qBeffara}} B_1$ is



or



$B_1 \xrightarrow{\text{Beffara}} B_2$ is



(up to duality)

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

  $B \xrightarrow{\text{qBeffara}} B_1$ is also a $B \xrightarrow{\text{Beffara}} B_1$

Characterization of $X \trianglelefteq B$

Theorem

The followings are equivalent:

1. $X \trianglelefteq B$
2. $B \xrightarrow{\text{qBeffara}}{}^{*} X$
3. $B \xrightarrow{\text{Beffara}}{}^{*} X$ *(up to iso)*

Characterization of $X \trianglelefteq B$

Theorem

The followings are equivalent:

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

$$P \; ::= \; X \;\mid\; P \otimes (N \parr P) \;\mid\; P \parr (N \otimes P)$$
$$N \; ::= \; X^{\perp} \;\mid\; N \otimes (P \parr N) \;\mid\; N \parr (P \otimes N)$$

Characterization of $X \trianglelefteq B$

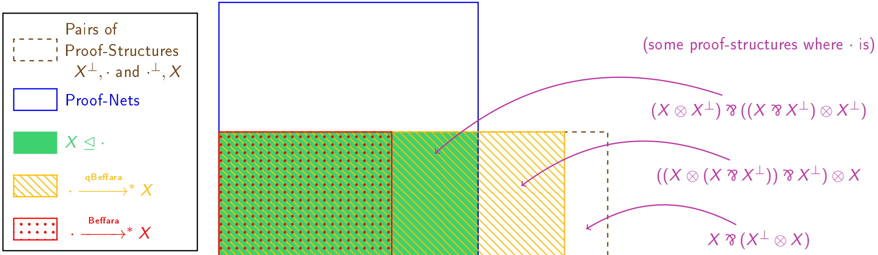
Theorem

The followings are equivalent:

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

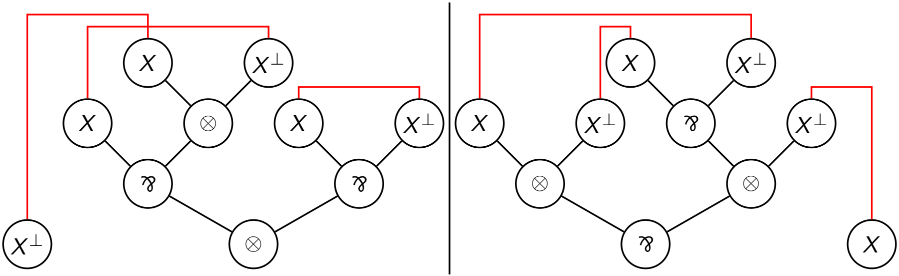
$$P ::= X \mid P \otimes (N \parr P) \mid P \parr (N \otimes P)$$
$$N ::= X^{\perp} \mid N \otimes (P \parr N) \mid N \parr (P \otimes N)$$

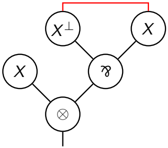
...but this is when looking at *formulas*! Looking at *proofs*, this is messier:



Retraction not generated by Beffara

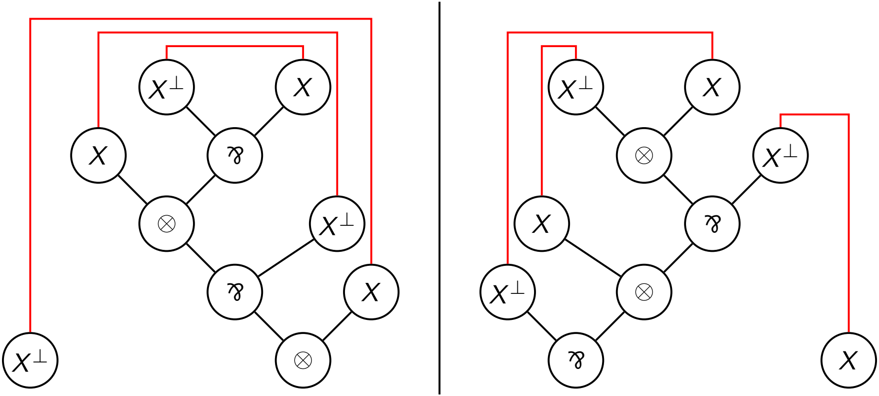
Proof of $X \lhd (X \otimes X^\perp) \parr ((X \parr X^\perp) \otimes X^\perp)$



Not generated by Beffara as no  in either proof-nets

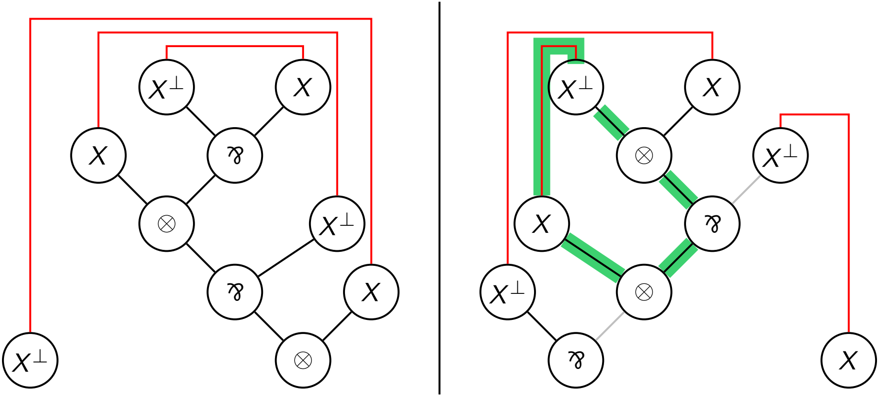
Incorrect retraction generated by Quasi-Beffara

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



Incorrect retraction generated by Quasi-Beffara

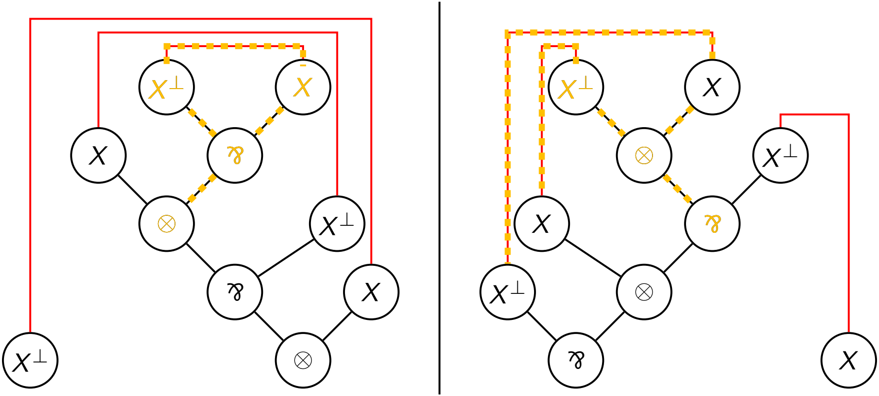
Not-Proof of $X \lhd ((X \otimes (X \parr X^\perp)) \parr X^\perp) \otimes X$



Incorrect

Incorrect retraction generated by Quasi-Beffara

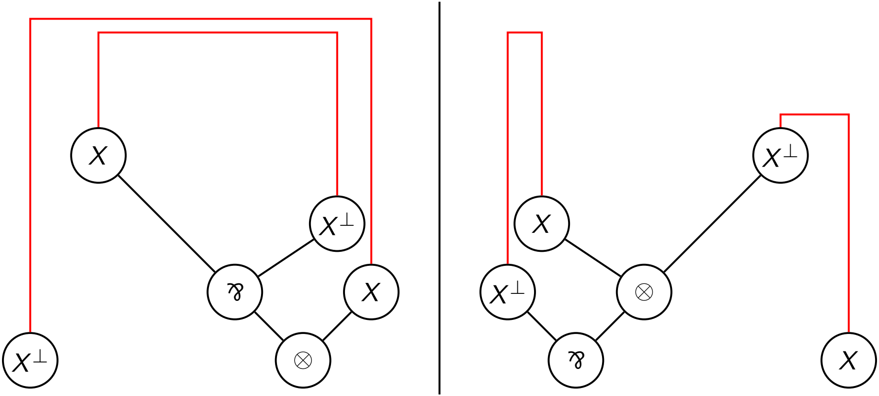
Not-Proof of $X \lhd ((X \otimes (X \parr X^\perp)) \parr X^\perp) \otimes X$



Can apply one step of Quasi-Beffara

Incorrect retraction generated by Quasi-Beffara

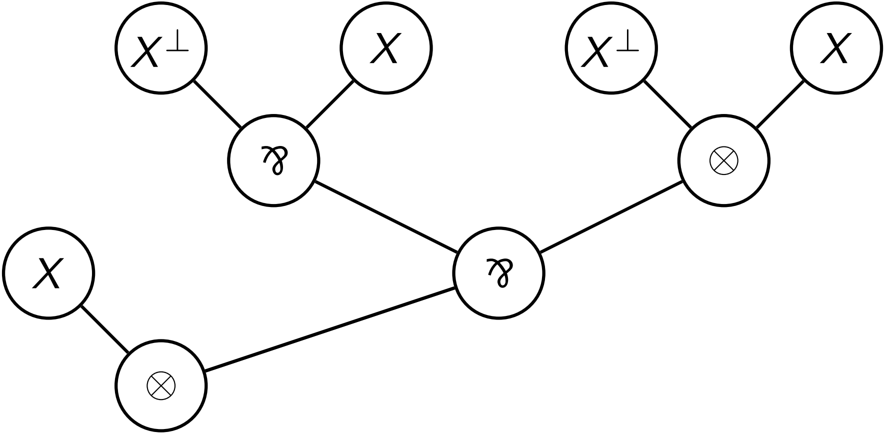
Not-Proof of $X \lhd ((X \otimes (X \parr X^{\perp})) \parr X^{\perp}) \otimes X$



This is Beffara, attainable from X by one step of Quasi-Beffara

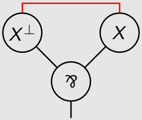
Formula not generated by Beffara without iso

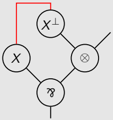
$$X \lhd X \otimes ((X^\perp ⅋ X) ⅋ (X^\perp \otimes X))$$



Generated by Beffara only up to isomorphism!

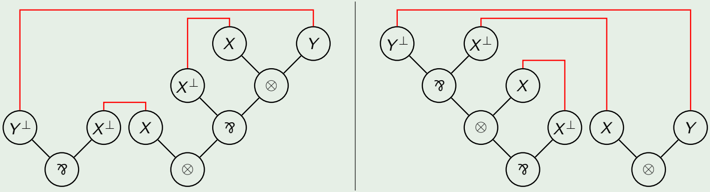
Generalization to $A \trianglelefteq B$?

Not only 

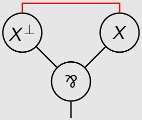
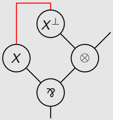
as a pattern, also 

; and maybe others?

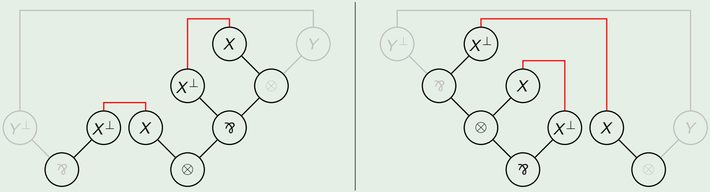
Example: $X \otimes Y \trianglelefteq X \otimes (X^\perp \parr (X \otimes Y))$



Generalization to $A \trianglelefteq B$?

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

Example: $X \otimes Y \trianglelefteq X \otimes (X^\perp \parr (X \otimes Y))$



Plan

What about the units? the *mix*$_2$-rule?

Result from [Balat and Di Cosmo, 1999]

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

$$\dfrac{\dfrac{}{\vdash 1}\,1}{\vdash \bot, 1}\,\bot$$

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

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

What about the units? the mix_2-rule?

Result from [Balat and Di Cosmo, 1999]

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

$$\dfrac{\dfrac{}{\vdash 1}\,1}{\vdash \bot, 1}\,\bot$$

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

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

The mix_2-rule does not matter: it is preserved by composition and the identity has none.

Retractions and Provability

Fact

$$!X \trianglelefteq !X \otimes !(X \otimes A) \iff \vdash A \text{ is provable}$$

$$X \trianglelefteq X \,\&\, (X \otimes A) \iff \vdash A \text{ is provable}$$

$$A \trianglelefteq A \oplus B \iff \vdash B^{\perp}, A \text{ is provable}$$

Retractions and Provability

Fact

$$!X \trianglelefteq\, !X \otimes !(X \otimes A) \iff \vdash A \textit{ is provable}$$

$$X \trianglelefteq X \,\&\, (X \otimes A) \iff \vdash A \textit{ is provable}$$

$$A \trianglelefteq A \oplus B \iff \vdash B^{\perp}, A \textit{ is provable}$$

Fragment	Provability
LL	Undecidable ☹
MELL	TOWER-hard ☹ (decidability is open)
MALL	PSPACE-complete ☹
ALL	P-complete

(an overview of these results on provability can be found in [Lincoln, 1995])

Conclusion

- $X \trianglelefteq B \iff B \xrightarrow{\text{Beffara}}{}^{*} X$ up to isomorphism
  with some subtleties on the proof-morphisms
- 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.

$X \lhd X \otimes (X^\perp \parr X)$ $\qquad$ $A \lhd A \otimes (X^\perp \parr X) \iff X \in A$

$!X \unlhd\ !X \otimes\ !(X \otimes A) \iff \vdash A$ $\qquad$ $X \unlhd X \,\&\, (X \otimes A) \iff \vdash A$

Thank you for your attention!

$A \unlhd A \,\&\, B \iff \vdash A^\perp, B$ $\qquad$ $A \unlhd A \oplus B \iff \vdash A, B^\perp$

$X \oplus Y \lhd ((X \oplus Z) \,\&\, (X \oplus Y)) \oplus Y$ $\qquad$ $?A \unlhd\ ??A$ $\qquad$ $?!A \unlhd\ ?!?!A$

References I

Balat, V. and Di Cosmo, R. (1999).
A linear logical view of linear type isomorphisms.
In Flum, J. and Rodríguez-Artalejo, M., editors, *Computer Science Logic*, volume 1683 of *Lecture Notes in Computer Science*, pages 250–265. Springer.

Di Guardia, R. and Laurent, O. (2023).
Type isomorphisms for multiplicative-additive linear logic.
In Gaboardi, M. and van Raamsdonk, F., editors, *International Conference on Formal Structures for Computation and Deduction (FSCD)*, volume 260 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:21. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik.

References II

Laurent, O. (2005).
Classical isomorphisms of types.
Mathematical Structures in Computer Science, 15(5):969–1004.

Lincoln, P. (1995).
Deciding provability of linear logic formulas.
In Girard, J.-Y., Lafont, Y., and Regnier, L., editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Note Series*, pages 109–122. Cambridge University Press.

Padovani, V. (2001).
Retracts in simple types.
In Abramsky, S., editor, *Typed Lambda Calculi and Applications*, volume 2044 of *Lecture Notes in Computer Science*, pages 376–384. Springer.

References III

Regnier, L. and Urzyczyn, P. (2002).
Retractions of types with many atoms.
http://arxiv.org/abs/cs/0212005.

Soloviev, S. (1983).
The category of finite sets and cartesian closed categories.
Journal of Soviet Mathematics, 22(3):1387–1400.

Back-up: What about other "simple" fragments?

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

$$?A \trianglelefteq ??A \qquad ?!A \trianglelefteq ?!?!A$$

Look like the only "basic" ones?

Back-up: What about other "simple" fragments?

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

$$?A \trianglelefteq ??A \qquad ?!A \trianglelefteq ?!?!A$$

Look like the only "basic" ones?

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

$$A \trianglelefteq A \,\&\, B \iff \vdash A^{\perp}, B \qquad \text{or} \qquad A \trianglelefteq A \oplus B \iff \vdash A, B^{\perp}$$

Retraction of an atom manageable.
But generally composition is bad due to the side condition:

$$X \oplus Y \lhd ((X \oplus Z) \,\&\, (X \oplus Y)) \oplus Y$$

comes from $X \oplus Y \lhd (X \oplus Y) \oplus Y$ without $\vdash X \oplus Z, (X \oplus Y)^{\perp}$

Back-up: What about other "simple" fragments?

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

$$?A \trianglelefteq ??A \qquad ?!A \trianglelefteq ?!?!A$$

  Look like the only "basic" ones?

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

$$A \trianglelefteq A \,\&\, B \iff \vdash A^\perp, B \qquad \text{or} \qquad A \trianglelefteq A \oplus B \iff \vdash A, B^\perp$$

  Retraction of an atom manageable.
  But generally composition is bad due to the side condition:

$$X \oplus Y \lhd ((X \oplus Z) \,\&\, (X \oplus Y)) \oplus Y$$

  comes from $X \oplus Y \lhd (X \oplus Y) \oplus Y$ without $\vdash X \oplus Z, (X \oplus Y)^\perp$

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