Nets between Determinism and Nondeterminism

Thesis dissertation

Paolo Tranquilli

ptranqui@pps.jussieu.fr

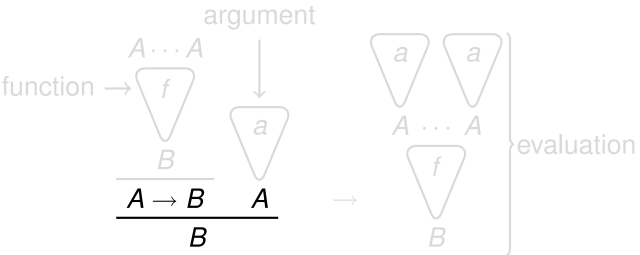
Dipartimento di Matematica
Università Roma Tre

Preuves Programmes et Systèmes
Université Paris Diderot

23 April 2009

Proofs as programs

Modus ponens, *cuts*, cut elimination = execution



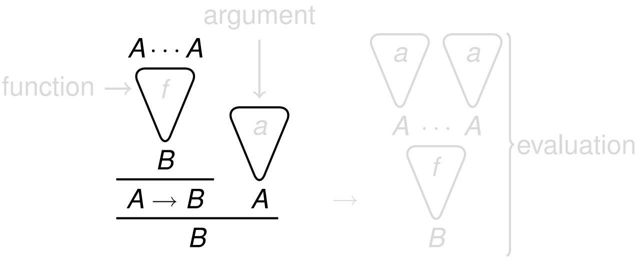
$$(\lambda x.f)\, a \to f[a/x]$$

Natural Deduction ⟷ λ-Calculus

William Alvin Howard.
The formulae-as-types notion of construction, volume to H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, pages 479–490.
Academic Press, 1980.

Proofs as programs

Modus ponens, cuts, cut elimination = execution



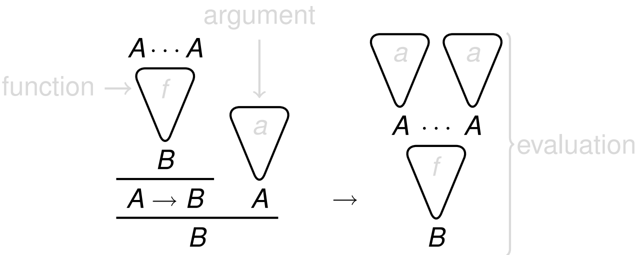
$$(\lambda x.f)\,a \to f[a/x]$$

Natural Deduction ⟷ λ-Calculus

William Alvin Howard.
The formulae-as-types notion of construction, volume to H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, pages 479–490.
Academic Press, 1980.

Proofs as programs

Modus ponens, cuts, cut elimination = execution



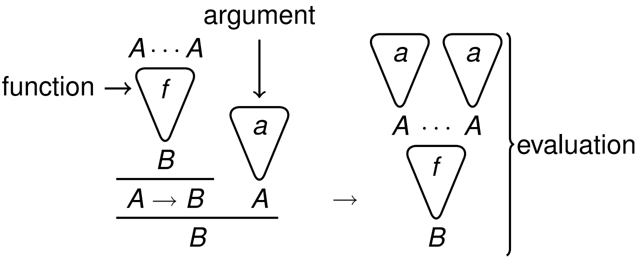
$(\lambda x.f)\, a \to f[a/x]$

Natural Deduction ⟷ λ-Calculus

William Alvin Howard.
The formulae-as-types notion of construction, volume to H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, pages 479–490.
Academic Press, 1980.

Proofs as programs

Modus ponens, cuts, cut elimination = execution



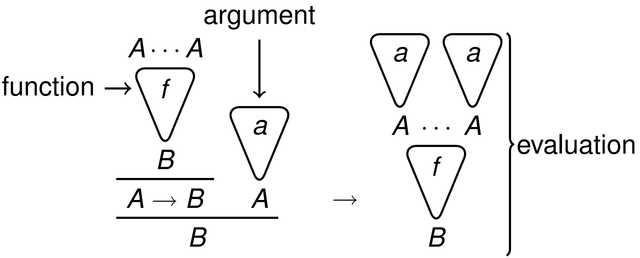
$(\lambda x.f)\, a \to f[a/x]$

Natural Deduction ⟷ λ-Calculus

William Alvin Howard.
The formulae-as-types notion of construction, volume to H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, pages 479–490.
Academic Press, 1980.

Proofs as programs

Modus ponens, cuts, cut elimination = execution



$$(\lambda x.f)\, a \to f[a/x]$$

Natural Deduction ⟷ λ-Calculus

William Alvin Howard.
The formulae-as-types notion of construction, volume to H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, pages 479–490.
Academic Press, 1980.

Denotational semantics

Aim: mathematical invariants of programs wrt reduction

1st example: Scott's domains

$$\text{type } A \mapsto [\![A]\!] \text{ topological space}$$

$$\text{program } t : A \to B \mapsto [\![t]\!] : [\![A]\!] \to [\![B]\!] \text{ continuous map}$$

$$\text{execution } t \to t' \mapsto [\![t]\!] = [\![t']\!] \text{ equality}$$

Dana Scott.
Continuous lattices.
In Lawvere, editor, *Toposes, Algebraic Geometry and Logic*, volume 274 of *Lecture Notes in Mathematics*, pages 97–136. Springer, 1972.

This approach has become an important tool in proof theory.

Linear Logic

Coherent spaces

reflexive graphs a semantics for system F (2^{nd} order λ-calculus)

$$\rightsquigarrow$$

$$A \to B = !A \multimap B$$

$$\rightsquigarrow$$

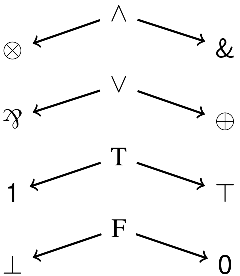
Linear Logic (LL)

! and its dual ? (the exponential modalities) control structural rules weakening and contraction

Jean-Yves Girard.
Linear logic.
Theoretical Computer Science, 50:1–102, 1987.

Multiplicatives and additives

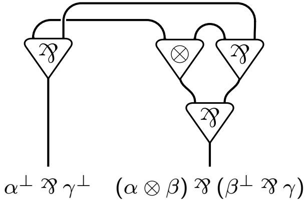
For lack of unrestricted structural rules, connectives and units are split



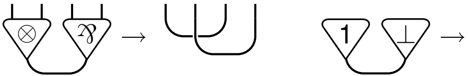
multiplicatives	additives
monoidal structure	cartesian structure

exponential isomorphism: $!A \otimes !B \cong !(A \,\&\, B)$

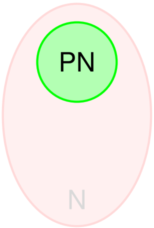
$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \alpha^\perp, \alpha}\,\mathsf{ax} \quad \dfrac{}{\vdash \beta, \beta^\perp}\,\mathsf{ax}}{\vdash \alpha^\perp, \alpha \otimes \beta, \beta^\perp}\,\otimes \quad \dfrac{}{\vdash \gamma, \gamma^\perp}\,\mathsf{ax}}{\vdash \alpha^\perp, \alpha \otimes \beta, \beta^\perp, \gamma, \gamma^\perp}\,\mathsf{mix}}{\vdash \alpha^\perp \parr \gamma^\perp, \alpha \otimes \beta, \beta^\perp, \gamma}\,\parr}{\vdash \alpha^\perp \parr \gamma^\perp, \alpha \otimes \beta, \beta^\perp \parr \gamma}\,\parr}{\vdash \alpha^\perp \parr \gamma^\perp, (\alpha \otimes \beta) \parr (\beta^\perp \otimes \gamma)}\,\parr
\quad \rightsquigarrow
$$



mix: naturally occurring iff $1 \cong \perp$ (e.g. in coherent spaces)
proof nets: desequentialized proof syntax with local cut elim.

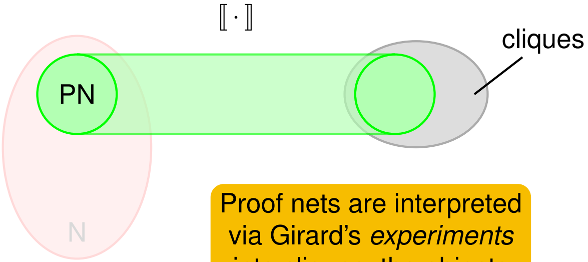


MLL and coherent spaces



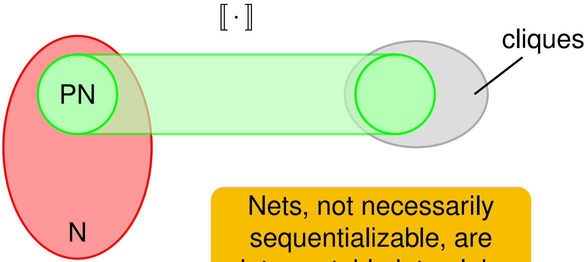
Proof-nets, corresponding to sequential proofs

MLL and coherent spaces



Proof nets are interpreted via Girard's *experiments* into cliques, the objects of coherent spaces

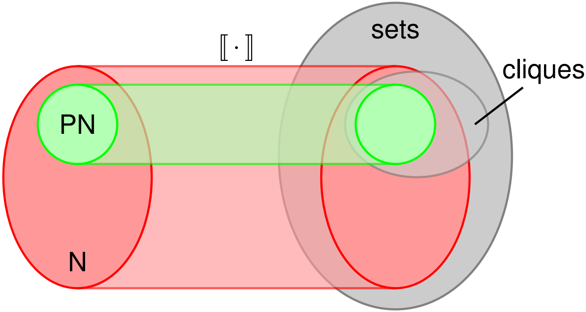
MLL and coherent spaces



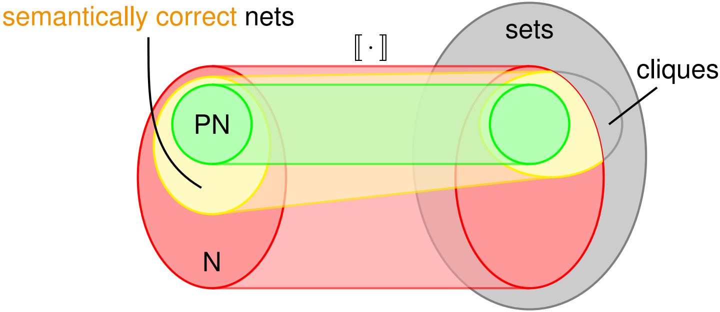
Nets, not necessarily sequentializable, are interpretable into plain *sets*

Proof nets are characterized among nets by a correctness criterion (absence of switching cycles)

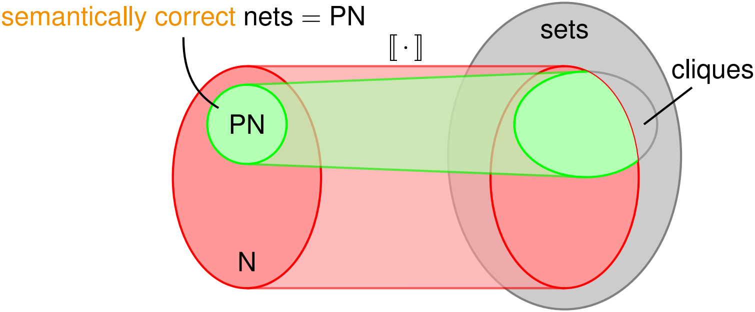
MLL and coherent spaces



MLL and coherent spaces



MLL and coherent spaces



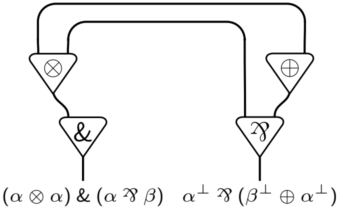
Christian Retoré.
A semantic characterisation of the correctness of a proof net.
Mathematical Structures in Computer Science, 7(5):445–452, October 1997.

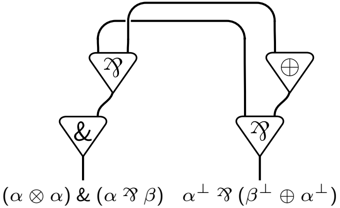
Proof nets

Multiplicative Additive LL (MALL)

Proofs as sets of multiplicative slices

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \alpha, \alpha^\perp} \qquad \dfrac{}{\vdash \alpha, \alpha^\perp}}{\vdash \alpha \otimes \alpha, \alpha^\perp, \alpha^\perp}}{\vdash \alpha \otimes \alpha, \beta^\perp \oplus \alpha^\perp, \alpha^\perp} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \alpha, \alpha^\perp} \qquad \dfrac{}{\vdash \beta, \beta^\perp}}{\vdash \alpha, \beta, \alpha^\perp, \beta^\perp}}{\vdash \alpha \parr \beta, \alpha^\perp, \beta^\perp}}{\vdash \alpha \parr \beta, \alpha^\perp, \beta^\perp \oplus \alpha^\perp}}{\dfrac{\vdash (\alpha \otimes \alpha) \,\&\, (\alpha \parr \beta), \alpha^\perp, \beta^\perp \oplus \alpha^\perp}{\vdash (\alpha \otimes \alpha) \,\&\, (\alpha \parr \beta), \alpha^\perp \parr (\beta^\perp \oplus \alpha^\perp)}} \rightsquigarrow
$$



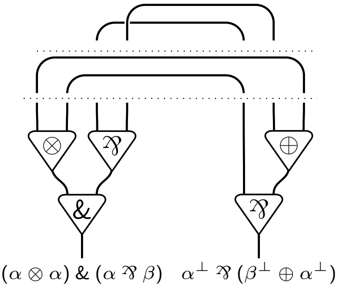


Proof nets

Multiplicative Additive LL (MALL)

Proofs as sets of multiplicative slices

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \alpha, \alpha^\perp} \qquad \dfrac{}{\vdash \alpha, \alpha^\perp}}{\vdash \alpha \otimes \alpha, \alpha^\perp, \alpha^\perp}}{\vdash \alpha \otimes \alpha, \beta^\perp \oplus \alpha^\perp, \alpha^\perp} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \alpha, \alpha^\perp} \qquad \dfrac{}{\vdash \beta, \beta^\perp}}{\vdash \alpha, \beta, \alpha^\perp, \beta^\perp}}{\vdash \alpha \parr \beta, \alpha^\perp, \beta^\perp}}{\vdash \alpha \parr \beta, \alpha^\perp, \beta^\perp \oplus \alpha^\perp}}{\dfrac{\vdash (\alpha \otimes \alpha) \mathbin{\&} (\alpha \parr \beta), \alpha^\perp, \beta^\perp \oplus \alpha^\perp}{\vdash (\alpha \otimes \alpha) \mathbin{\&} (\alpha \parr \beta), \alpha^\perp \parr (\beta^\perp \oplus \alpha^\perp)}} \rightsquigarrow$$



Dominic Hughes and Rob van Glabbeek.
Proof nets for unit-free multiplicative-additive linear logic.
In *LICS*, pages 1–10. IEEE Computer Society Press, 2003.

Correctness via absence of certain unions of switching cycles in the superposition with jumps.

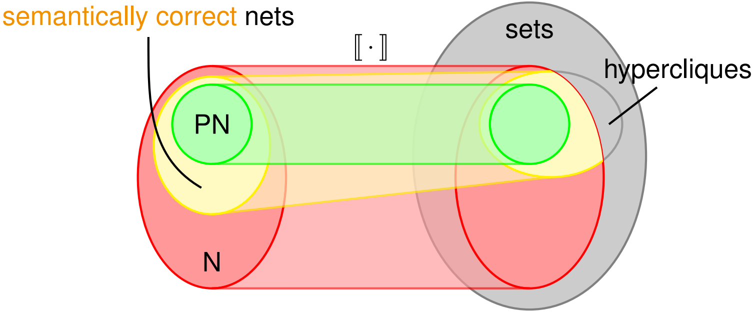
MALL and models of LL

- Coherent spaces are too permissive for MALL...
- ...just like they are for PCF (extension of λ-calculus with constants)
- research on PCF led to the strongly stable model, which in turn led to hypercoherent spaces for LL.
- switch from *graphs* (coherent sp.) to *hypergraphs*.

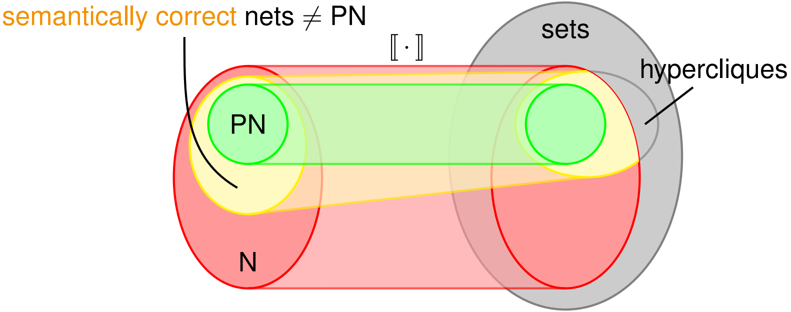
Antonio Bucciarelli and Thomas Ehrhard.
Sequentiality and strong stability.
In *LICS*. IEEE Computer Society Press, 1991.

Thomas Ehrhard.
Hypercoherence: A strongly stable model of linear logic.
In *Advances in Linear Logic*, pages 83–108. Cambridge University Press, 1995.

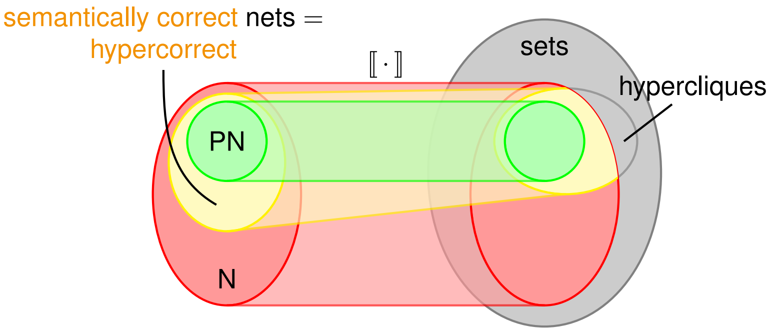
MALL and hypercoherent spaces



MALL and hypercoherent spaces



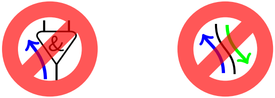
MALL and hypercoherent spaces



Paolo Tranquilli.
A characterization of hypercoherent semantic correctness in multiplicative additive linear logic.
In Michael Kaminski and Simone Martini, editors, *CSL*, volume 5213 of *Lecture Notes in Computer Science*, pages 246–261. Springer, 2008.

Hypercorrectness

- a criterion with paths
- wrt HvG usual proof nets, restriction of switching paths and unions to be considered. Oriented paths.



- stable under reduction
- implied by correctness (proves hypercoherent spaces are a model for proof nets)

Conjecture

On switching connected structures correctness and hypercorrectness coincide, i.e. correctness and semantic correctness coincide.

Switching connectedness is associated with absence of mix.

Linearity in Logic and Computer Science

Coherent spaces are vaguely vector spaces with product. . .

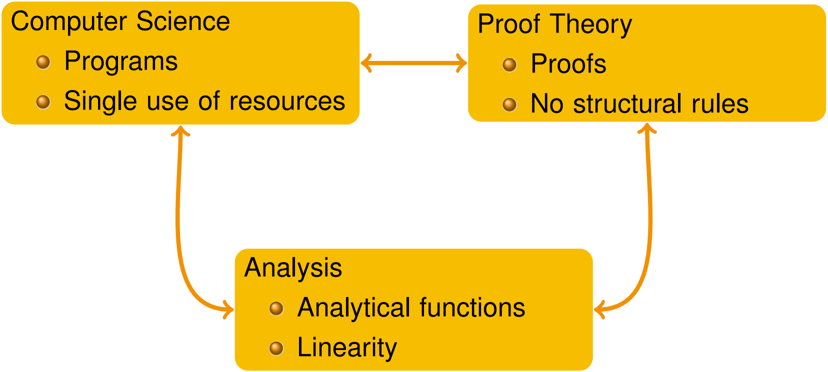
Computer Science
- Programs
- Single use of resources

⟷

Proof Theory
- Proofs
- No structural rules

Linearity in Logic and Computer Science

Coherent spaces are vaguely vector spaces with product. . .



From coherent spaces to finiteness spaces

- Coherent spaces: interaction $u \perp v \iff \# u \cap v \leq 1$.
- Finitenss spaces: $u \perp v \iff \# u \cap v < \omega$ finite.
- Correspond to (some) vector spaces with linear topology.
- Have the derivation operation

$$\rightsquigarrow$$

$$A \to B = !A \multimap B = \text{analytical functions}$$

$$\rightsquigarrow$$

Differential Linear Logic (DiLL)

Thomas Ehrhard.
Finiteness spaces.
Mathematical. Structures in Comp. Sci., 15(4):615–646, 2005.

Thomas Ehrhard and Laurent Regnier.
Differential interaction nets.
Theor. Comput. Sci., 364(2):166–195, 2006.

Linearity in Logic and Computer Science

Computer Science

- Programs
- Single use of resources

Proof Theory

- Proofs
- No structural rules

Analysis

- Analytical functions
- Linearity
- Sums
- Differentials: linear aproximation

Linearity in Logic and Computer Science

Computer Science

- Programs
- Single use of resources
- Nondeterminism
- Depletable resources

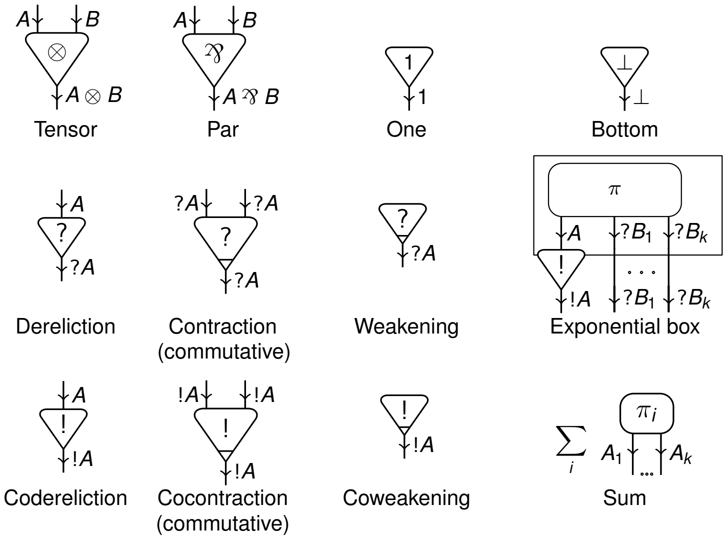
Proof Theory

- Proofs
- No structural rules
- Costructural rules, sum ($\& = \oplus$)
- Codereliction

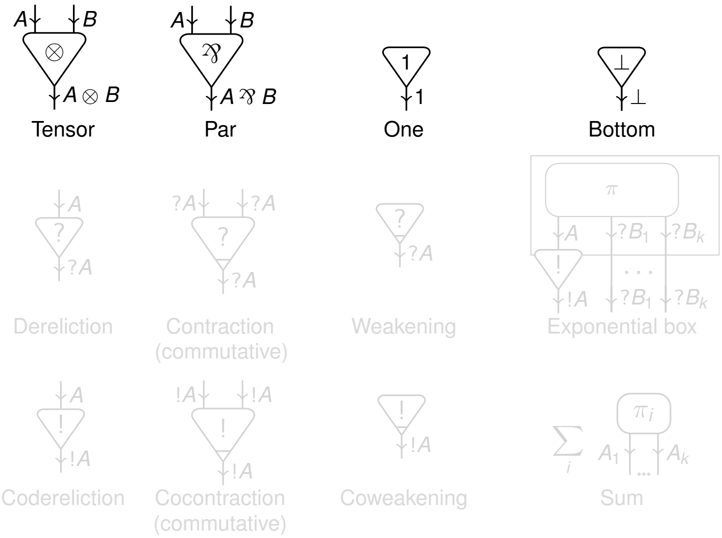
Analysis

- Analytical functions
- Linearity
- Sums
- Differentials: linear aproximation

The nets: Family picture

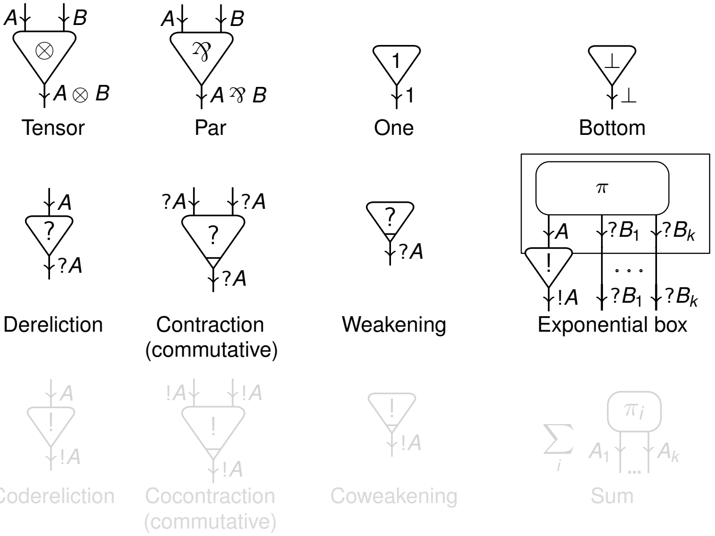


The nets: MLL

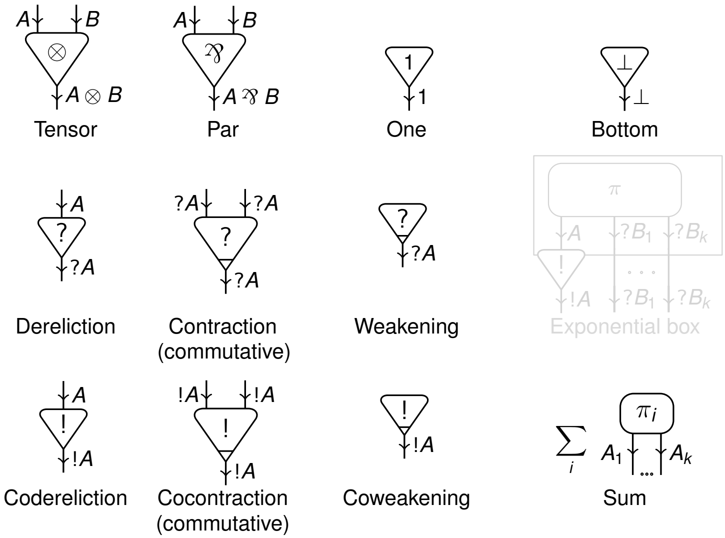


A B $A \otimes B$
Tensor
A B $A ⅋ B$
Par
1
One
$\bot$
Bottom
A ? $?A$
Dereliction
$?A$ $?A$? $?A$
Contraction (commutative)
? $?A$
Weakening
π A $?B_1$ $?B_k$! $!A$ $?B_1$ $?B_k$
Exponential box
A ! $!A$
Codereliction
$!A$ $!A$! $!A$
Cocontraction (commutative)
! $!A$
Coweakening
$\sum_i$ π_i A_1 A_k
Sum

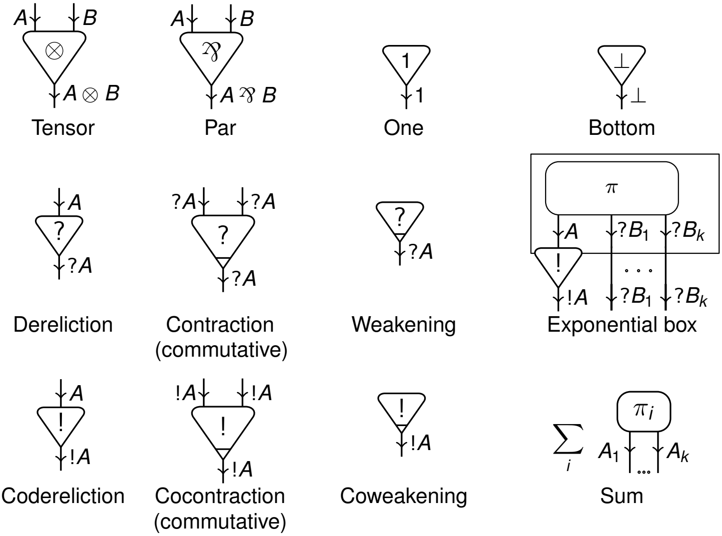
The nets: Multiplicative Exponential LL (MELL)



The nets: Differential Interaction Nets



The nets: Differential Nets (DiLL)

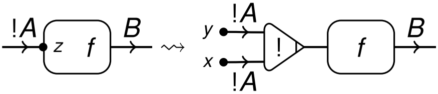


Cocontraction and coweakening

Cocontraction

Analysis: $f(x+y)$ out of $f(z)$.

Nets:

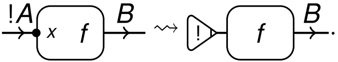


Programs: joining resources.

Coweakening

Analysis: Evaluation in 0.

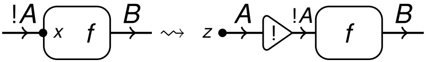
Nets:



Programs: Empty resource.

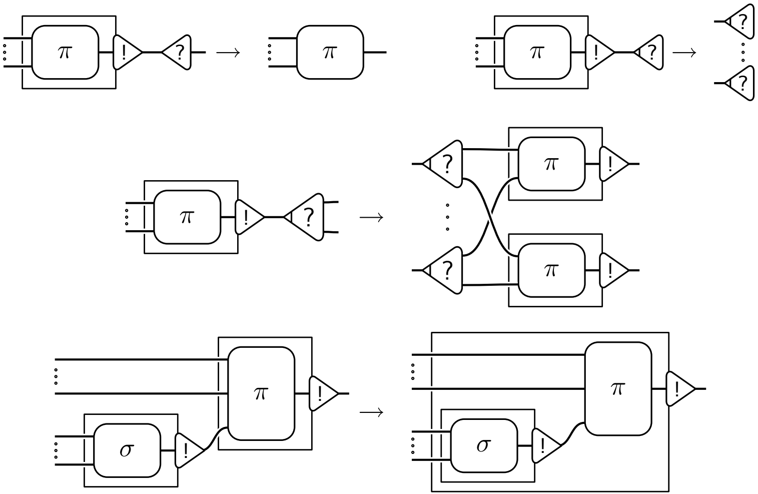
Codereliction

Analysis: Derivative in 0, giving a linear function: $\left.\dfrac{\partial f(x)}{\partial x}\right|_{x=0} \cdot z.$

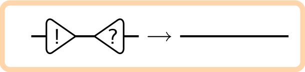
Nets: 

Programs: a single use resource.

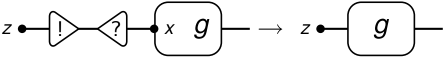
The known reductions



Codereliction vs dereliction

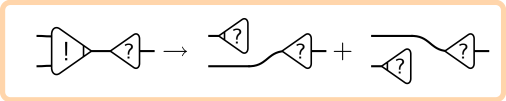


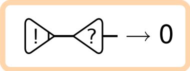
Analysis: $\dfrac{\partial g \cdot x}{\partial x} = g \implies \left.\dfrac{\partial g \cdot x}{\partial x}\right|_{x=0} \cdot z = g \cdot z$:



Programs: a single-use query encounters a single-use resource and is resolved.

Cocontranction and coweakening vs dereliction





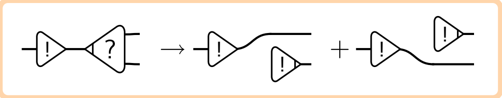
Analysis: linearity!

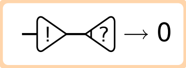
$$g \cdot (x + y) = g \cdot x + g \cdot y, \qquad g \cdot 0 = 0.$$

Programs: a single-use query presented with two resources $\rightsquigarrow$ nondeterministic choice.
Or presented with an empty one $\rightsquigarrow$ nondeterministic dead end.

Codereliction vs contraction and coweakening

Completely simmetric!





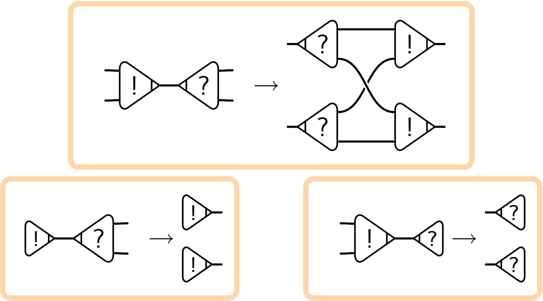
Analysis: laws of derivation!

$$\frac{\partial f(x,x)}{\partial x}\Big|_{x=0} = \left(\frac{\partial f(y,z)}{\partial y}, \frac{\partial f(y,z)}{\partial z}\right)\Big|_{y,z=x} \cdot \frac{\partial(x,x)}{\partial x}\Big|_{x=0} = \frac{\partial f(y,0)}{\partial y}\Big|_{y=0} + \frac{\partial f(0,z)}{\partial z}\Big|_{z=0}$$

and $\frac{\partial k}{\partial x} = 0$.

Programs: two queries meets a single-use resource $\rightsquigarrow$ nondeterministic choice.
Or an empty query meets a single use resource $\rightsquigarrow$ an error (not affine).

Routing: (co)contractions and (co)weakenings

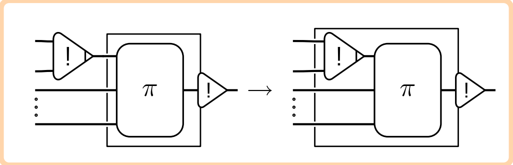


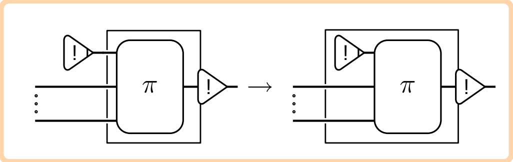
Analysis: Commutation of sum and sharing.

Programs: routing of queries and resources.

Cocontraction and coweakening vs box

Recall that cocontractions and coweakenings are, in a way, boxes.

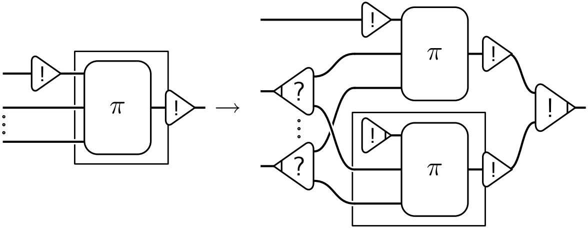




analysis: plain composition.

programs: operations on resources transported inside a package.

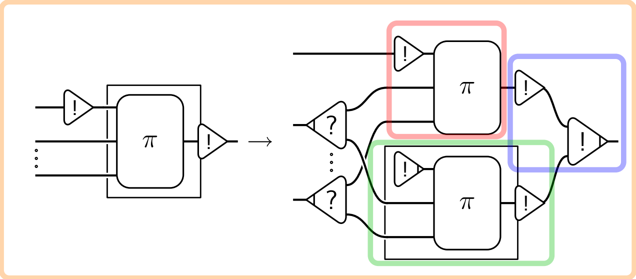
Pandora's box: codereliction vs box



Analysis: $\frac{\partial f(g(x))}{\partial x}\Big|_{x=0} \cdot Z = \frac{\partial f(y)}{\partial y}\Big|_{y=g(0)} \cdot \frac{\partial g(x)}{\partial x}\Big|_{x=0} \cdot Z$

Programs: reusable package gets a single-use resource $\rightsquigarrow$ a single-use copy gets the resource, others an empty one.

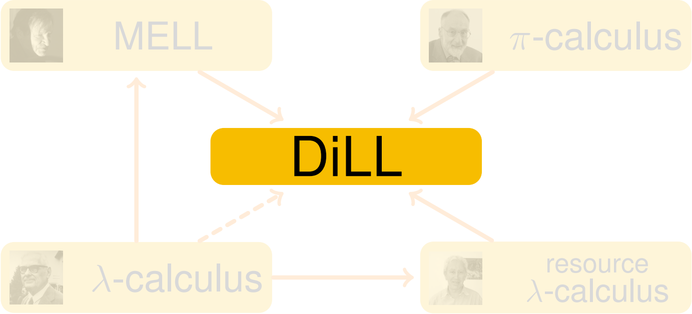
Pandora's box: codereliction vs box



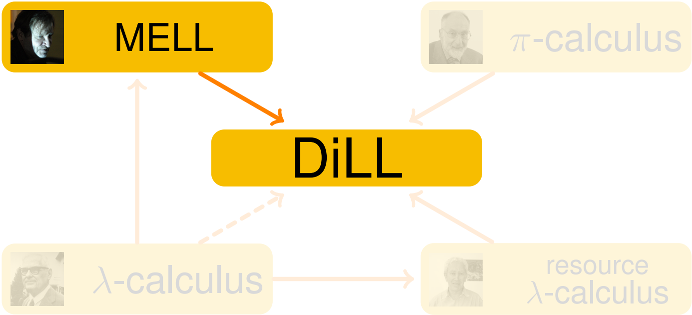
Analysis: $\left.\frac{\partial f(g(x))}{\partial x}\right|_{x=0} \cdot Z = \left.\frac{\partial f(y)}{\partial y}\right|_{y=g(0)} \cdot \left.\frac{\partial g(x)}{\partial x}\right|_{x=0} \cdot Z$

Programs: reusable package gets a single-use resource $\rightsquigarrow$ a single-use copy gets the resource, others an empty one.

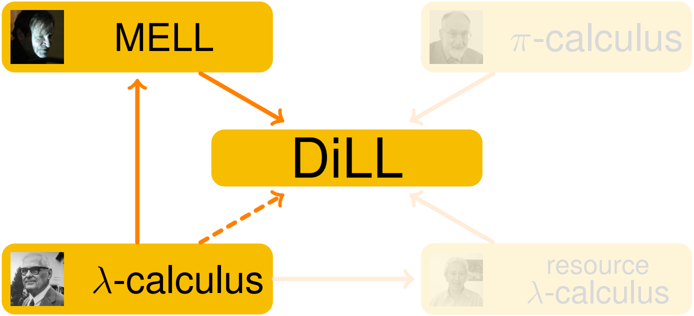
Quite an expressive system



Quite an expressive system



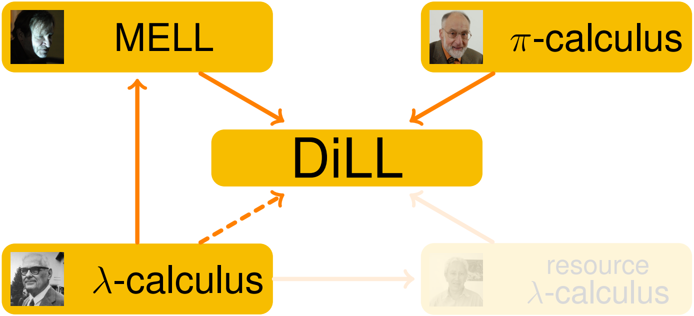
Quite an expressive system



Vincent Danos.
La Logique Linéaire appliquée à l'étude de divers processus de normalisation (principalement du λ-calcul).
Thèse de doctorat, Université Paris VII, 1990.

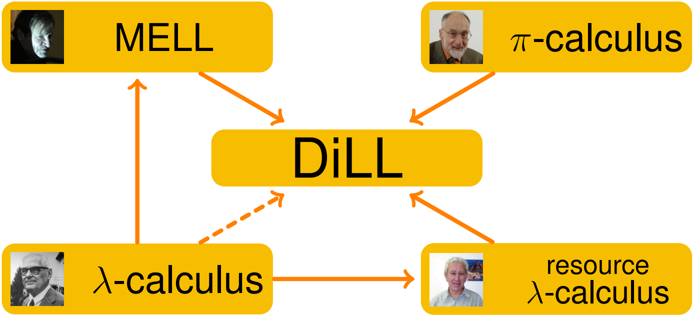
Laurent Regnier.
Lambda-Calcul et Réseaux.
Thèse de doctorat, Université Paris VII, 1992.

Quite an expressive system



Thomas Ehrhard and Olivier Laurent.
Interpreting a finitary pi-calculus in differential interaction nets.
CONCUR, *LNCS* vol. 4703, pages 333–348, 2007.

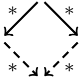
Quite an expressive system



Gérard Boudol.
The lambda-calculus with multiplicities.
INRIA Research Report 2025, 1993.

Paolo Tranquilli.
Intuitionistic differential nets and lambda calculus.
To appear on *Theoretical Computer Science*, 2008.

What do we want?

- Confluence: 
- Termination: no infinite reduction $\to \to \cdots \to \cdots$
- Assure a unique result regardless of reduction strategy.
- Normal forms form a category.

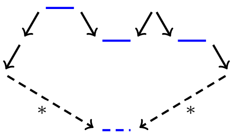
However in DiLL:

- confluence fails without associativity of (co)contractions;
- we introduce it as an equivalence, together (while we are at it) with other ones:
  - commutation of contractions with box borders,
  - a version of the exponential isomorphism, $!(\pi + \sigma) \sim !(\pi) \cdot !(\sigma)$.

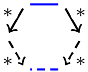
Reduction modulo equivalence

- For confluence, most important property is Church Rosser modulo.
  CR$\sim$ holds if

$$(\to \cup \leftarrow \cup \sim)^* \subseteq \xrightarrow{*}\sim\xleftarrow{*}, \quad \text{for ex.}$$



Not equivalent to confluence modulo:



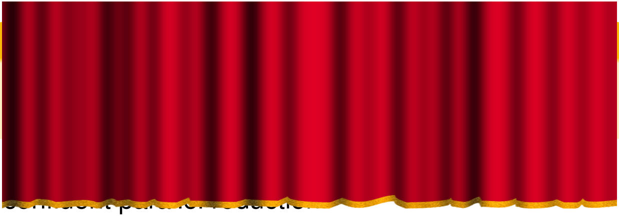
- $t \in A$ is SN$\sim$ if it is SN for $\sim\to\sim$.

Enno Ohlebusch.
Church-Rosser theorems for abstract reduction modulo an equivalence relation.
In *RTA*, *LNCS* vol. 1379, pages 17–31, 1998.

Confluence

Theorem

Reduction of switching acyclic untyped DiLL*nets is CR modulo* $\sim$.



Michele Pagani and Lorenzo Tortora de Falco.
Strong normalization property for second order linear logic.
To appear on *Theor. Comput. Sci.*, 2008.

Paolo Tranquilli.
Confluence of pure differential nets with promotion.
Submitted to *Computer Science Logic*, 2008.

Confluence

Theorem

Reduction of switching acyclic untyped DiLL*nets is CR modulo* $\sim$.

Theorem (Finite developments)

Reduction not reducing "new" cuts is strongly normalizing modulo $\sim$.

Local confluence of developments $\Longrightarrow$ strongly confluent parallel reduction.

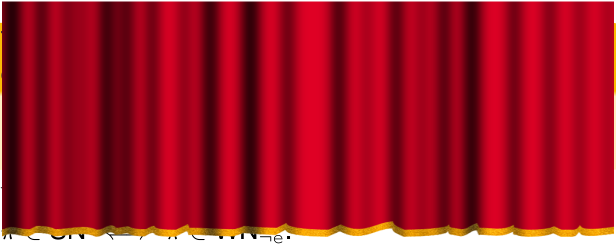
Michele Pagani and Lorenzo Tortora de Falco.
Strong normalization property for second order linear logic.
To appear on *Theor. Comput. Sci.*, 2008.

Paolo Tranquilli.
Confluence of pure differential nets with promotion.
Submitted to *Computer Science Logic*, 2008.

Termination

Theorem

Reduction is strongly normalizing in the simply typed case.



Michele Pagani and Lorenzo Tortora de Falco.
Strong normalization property for second order linear logic.
To appear on *Theor. Comput. Sci.*, 2008.

Michele Pagani.
The cut-elimination theorem for differential nets with boxes.
Accepted in *Typed Lambda Calculi and Applications*, 2009.

Termination

Theorem

Reduction is strongly normalizing in the simply typed case.

Theorem (standardization, or striction, or conservation, with Pagani)

For π *laxly typed sw. acyclic* DiLL *net,*
if $\pi \xrightarrow{\neg\mathtt{er}} \pi'$*, then* $\pi' \in \mathsf{SN}_{\neg\mathtt{er}} \implies \pi \in \mathsf{SN}_{\neg\mathtt{er}}$.

$\xrightarrow{\neg\mathtt{e}}$ stands for non erasing reduction. It follows that $\pi \in \mathsf{SN} \iff \pi \in \mathsf{WN}_{\neg\mathtt{e}}$.

Michele Pagani and Lorenzo Tortora de Falco.
Strong normalization property for second order linear logic.
To appear on *Theor. Comput. Sci.*, 2008.

Michele Pagani.
The cut-elimination theorem for differential nets with boxes.
Accepted in *Typed Lambda Calculi and Applications*, 2009.

Full resource calculus

$$t ::= x \mid \lambda x.t \mid \langle t \rangle\, B$$

with bags $B = t_1^{e_1} \cdots t_k^{e_k}$, $e_k \leq \infty$.

A calculus for nondeterminism with a structurecontextual reduction, borrowed from differential λ-calculus.

$$\langle \lambda x.t \rangle\, u \cdot B \to \langle \lambda x.\tfrac{\partial t}{\partial x} \cdots u \rangle\, B, \qquad \langle \lambda x.t \rangle\, u^{\infty} \cdot B \to \langle \lambda x.t[x + u/x] \rangle\, B.$$

The target of Taylor expansion of λ-calculus.

Gérard Boudol.
The lambda-calculus with multiplicities.
1993.

Thomas Ehrhard and Laurent Regnier.
The differential lambda-calculus.
Theor. Comput. Sci., 309(1):1–41, 2003.

Thomas Ehrhard and Laurent Regnier.
Uniformity and the Taylor expansion of ordinary lambda-terms.
Theor. Comput. Sci., 403(2-3):347–372, 2008.

Resource calculus and differential nets

Modular translation $t \mapsto t^\circ$.

Theorem (Simulation)

- $s \to t \iff s^\circ \xrightarrow{\mathtt{m}}\xrightarrow{\mathtt{e}*} t^\circ$,
- $s \xrightarrow{*} t \iff s^\circ \xrightarrow{*} t^\circ$

(modulo $\sim$)

Corollary (Confluence (and termination))

Full resource calculus is confluent (and normalizing in the typed case).

Vincent Danos.
La Logique Linéaire appliquée à l'étude de divers processus de normalisation (principalement du λ-calcul).
Thèse de doctorat, Université Paris VII, 1990.

Laurent Regnier.
Lambda-Calcul et Réseaux.
Thèse de doctorat, Université Paris VII, 1992.

Paolo Tranquilli.
Intuitionistic differential nets and lambda calculus.
To appear on *Theoretical Computer Science*, 2008.

Thanks

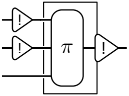
Nondeterminism and confluence?

Q: How come we speak of confluence with nondeterminism around?

A: confluence ensures nondeterministic choice is not triggered by what we reduce.

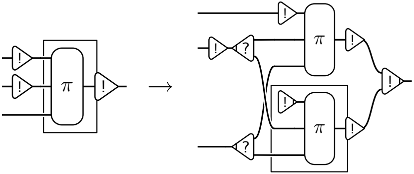
Reducing two different redexes gives the same set of nondeterministic choices in the end.

We need associativity and neutrality



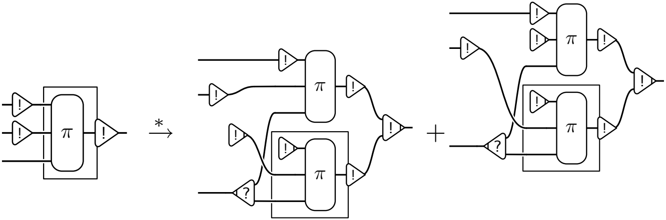
- Contractions and cocontractions need to be swapped to have confluence (associativity).
- Other confluence diagrams require merging (co)weakenings with (co)contractions (neutrality).

We need associativity and neutrality



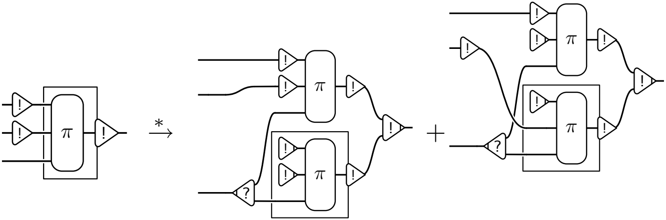
- Contractions and cocontractions need to be swapped to have confluence (associativity).
- Other confluence diagrams require merging (co)weakenings with (co)contractions (neutrality).

We need associativity and neutrality



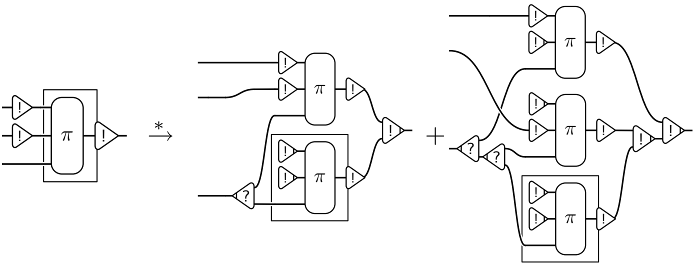
- Contractions and cocontractions need to be swapped to have confluence (associativity).
- Other confluence diagrams require merging (co)weakenings with (co)contractions (neutrality).

We need associativity and neutrality



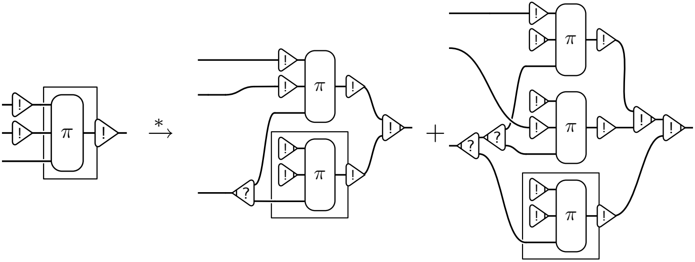
- Contractions and cocontractions need to be swapped to have confluence (associativity).
- Other confluence diagrams require merging (co)weakenings with (co)contractions (neutrality).

We need associativity and neutrality



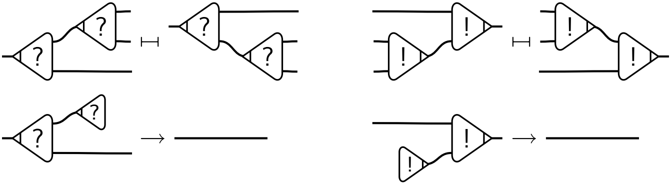
- Contractions and cocontractions need to be swapped to have confluence (associativity).
- Other confluence diagrams require merging (co)weakenings with (co)contractions (neutrality).

We need associativity and neutrality



- Contractions and cocontractions need to be swapped to have confluence (associativity).
- Other confluence diagrams require merging (co)weakenings with (co)contractions (neutrality).

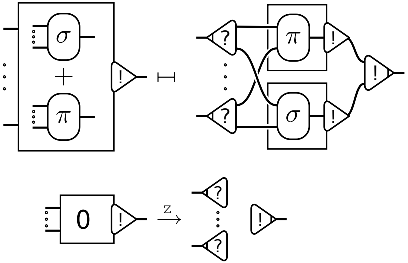
Associative equivalence and neutral reduction



- associative equivalence $\sim = \mapsto^*$;
- neutral reduction (cannot be reversed).

Compulsory!

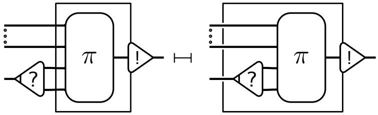
Bang sum equivalence and bang zero reduction

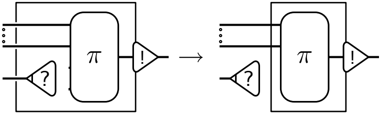


with $\sigma, \pi \neq 0$,

A reusable nondeterministic resource is equivalent to joining the various reusable resources.

Push equivalence and pull reduction

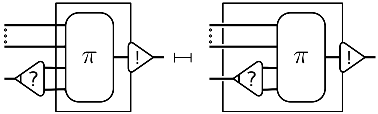


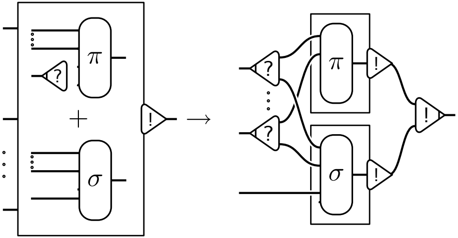


with $\pi \neq 0$.

These latter equivalences and reductions are optional (but inseparable).

Push equivalence and pull reduction

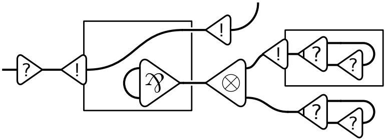




with $\pi \neq 0$.

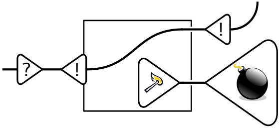
These latter equivalences and reductions are optional
(but inseparable).

“Lazarus” clashes



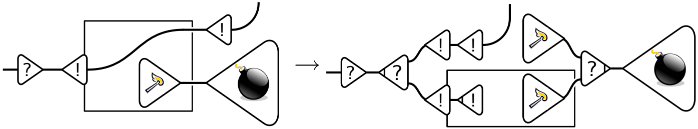
In LL Lazarus clashes do not negate standardization.

“Lazarus” clashes



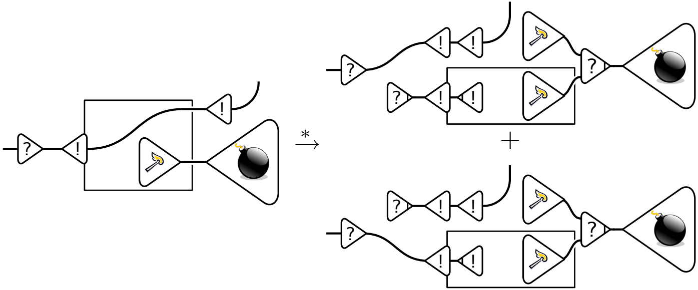
In LL Lazarus clashes do not negate standardization.

"Lazarus" clashes



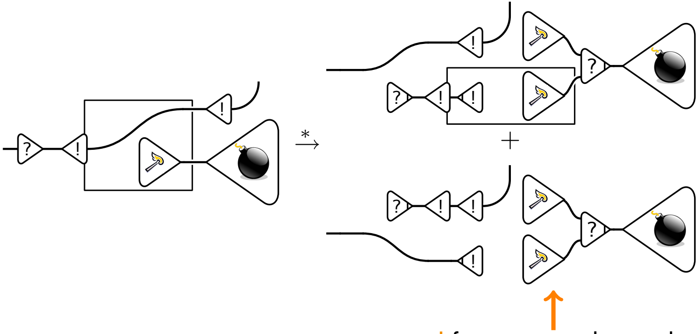
In LL Lazarus clashes do not negate standardization.

“Lazarus” clashes



In LL Lazarus clashes do not negate standardization.

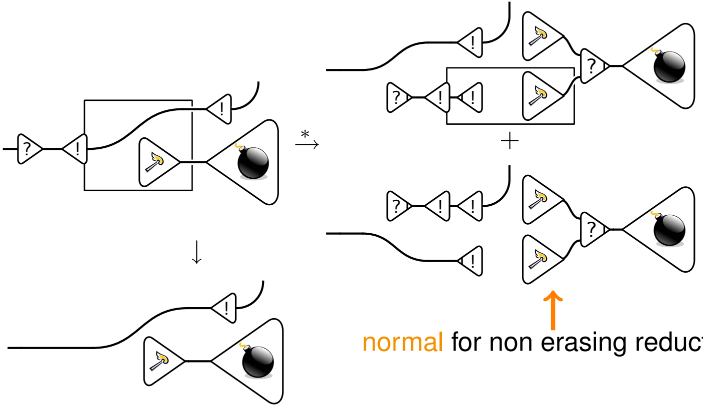
“Lazarus” clashes



normal for non erasing reduction

In LL Lazarus clashes do not negate standardization.

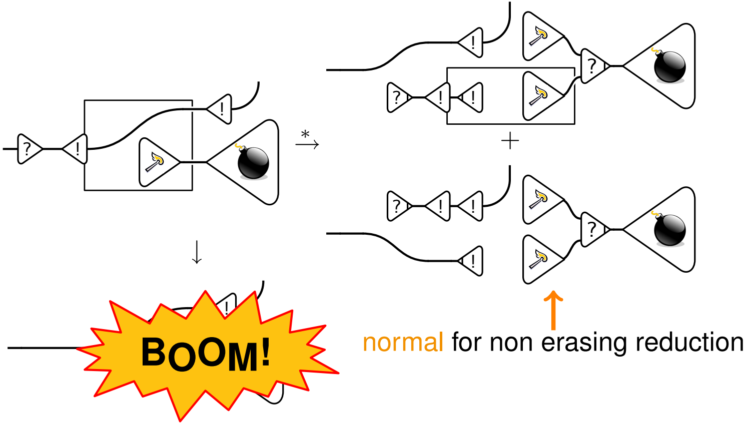
“Lazarus” clashes



normal for non erasing reduction

In LL Lazarus clashes do not negate standardization.

"Lazarus" clashes



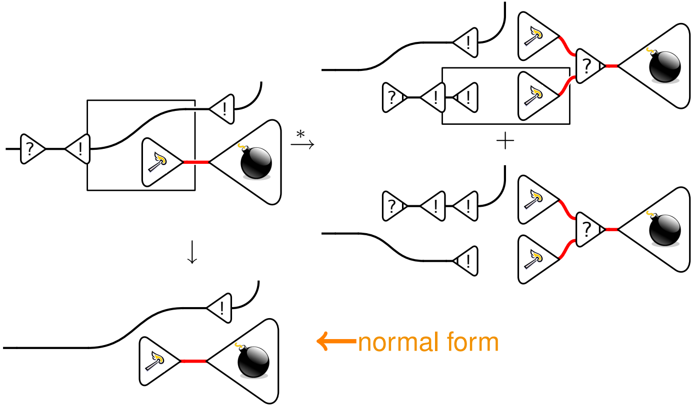
In LL Lazarus clashes do not negate standardization.

Lax typing

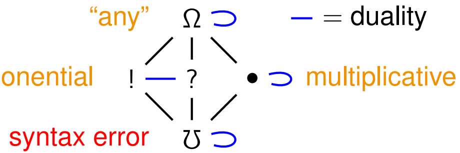
- We introuduce a very weak form of typing.
- It does not disallow nets, but assigns a special syntax error type to exponential clashes.
- We block the reduction of “syntax errror”-typed cuts.
- Any typing system avoiding clashes can be embedded in this one.

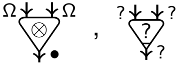
Details

“Lazarus” clashes



Lax typing

- 5 ordered types: 

- expected typing rules, for example 

- All type mismatches are allowed, but resulting type is the inf of the two.
- ℧-typed cuts are blocked.
- Types are preserved during reduction.

Lax typing does not disallow nets, but provides a mechanism to annotate exponential clashes and prevent them from resurrecting.

Back