Differential Linear Logic and Polarization

Speaker: Michele Pagani

Lionel Vaux
lionel.vaux@univ-savoie.fr

Laboratoire de Mathématiques
Université de Savoie, France

Typed Lambda Calculi and Applications, 2009, Brasilia

Where is the author ?

Taking care of his newborn son:

Ulysse, born on June the 11th

Merci Michele !

\tableofcontents

Context

Cut elimination = computation

Curry–Howard correspondence

Context

λ-calculus

Functional programming

Intuitionistic logic

Cut elimination = computation

Curry–Howard correspondence

Context

Functional programming

λ-calculus

Intuitionistic logic

Girard's Linear logic
$!A \multimap B$

Coherence semantics

Curry–Howard correspondence

Denotational semantics

Context

λ-calculus

Functional programming

Intuitionistic logic

Linear logic

Coherence semantics

Parigot 1992
$\lambda\mu$-calculus

Griffin 1990
Control

Classical logic

Polarized linear logic
Laurent 2002

Correlation spaces
Girard 1991

Curry–Howard correspondence

Context

Functional programming

λ-calculus

Intuitionistic logic

Linear logic

Coherence semantics

$\lambda\mu$-calculus

Control

Classical logic

Polarized linear logic

Correlation spaces

Curry–Howard correspondence

Differential Linear Logic

Quantitative semantics *Girard,* predates coherence

Context

λ-calculus

Functional programming

Intuitionistic logic

Linear logic

Coherence semantics

$\lambda\mu$-calculus

Control

Classical logic

Polarized linear logic

Correlation spaces

Curry–Howard correspondence

Ehrhard Laurent 2007
Concurrency

Ehrhard Regnier 2003
Differential λ-calculus

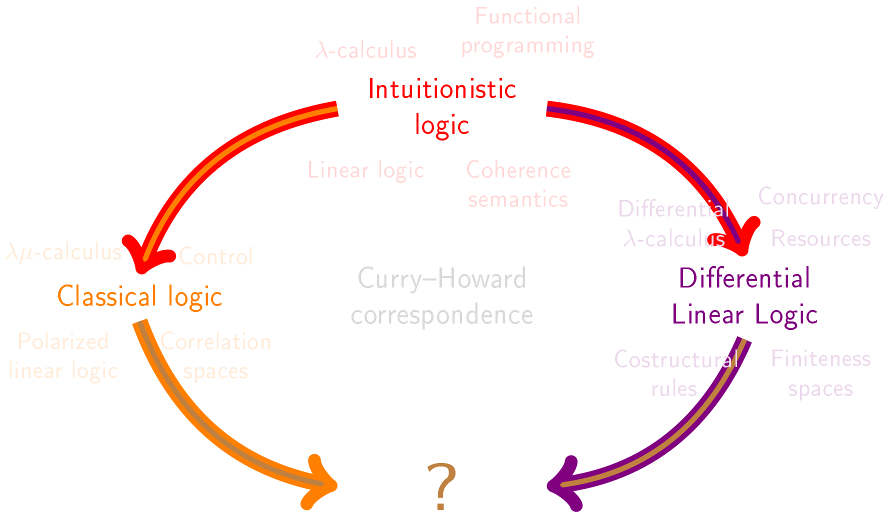
Ehrhard Regnier 2006
Resources

Differential Linear Logic

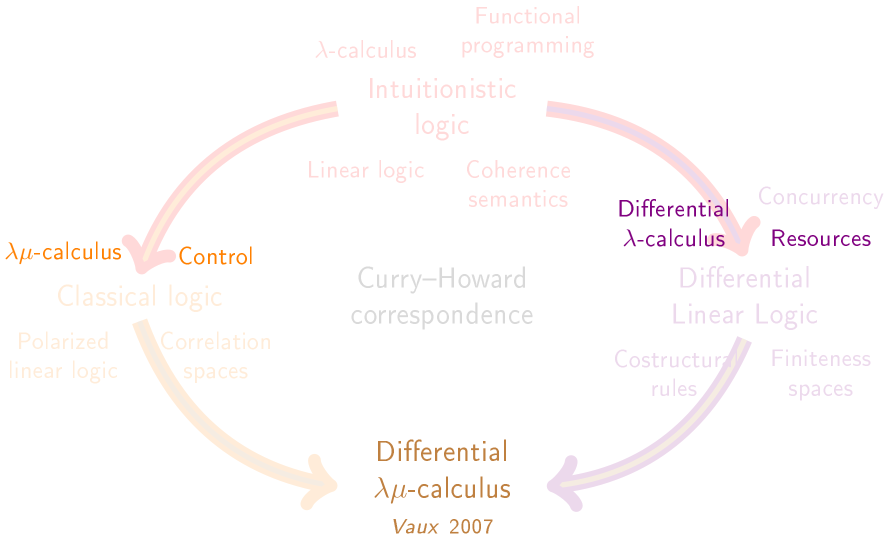
Costructural rules
Ehrhard Regnier 2006

Finiteness spaces
Ehrhard 2001

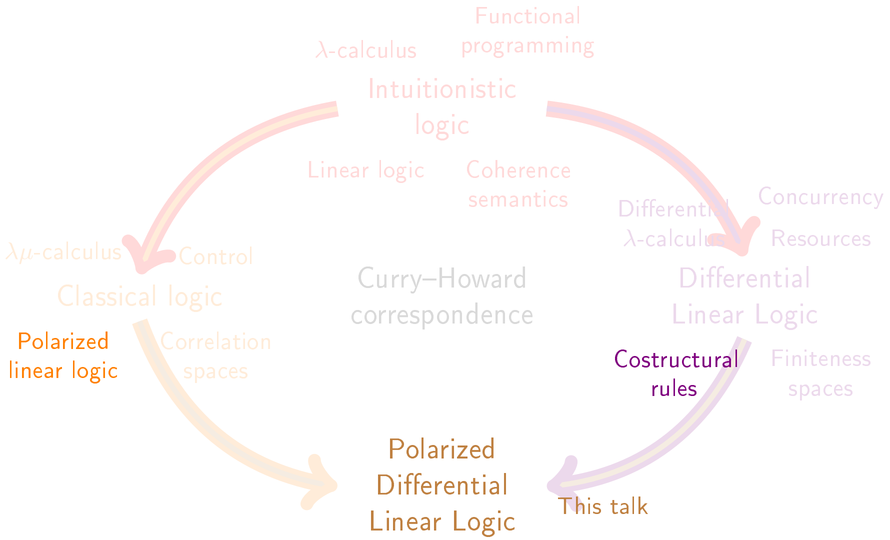
Context, Motivations



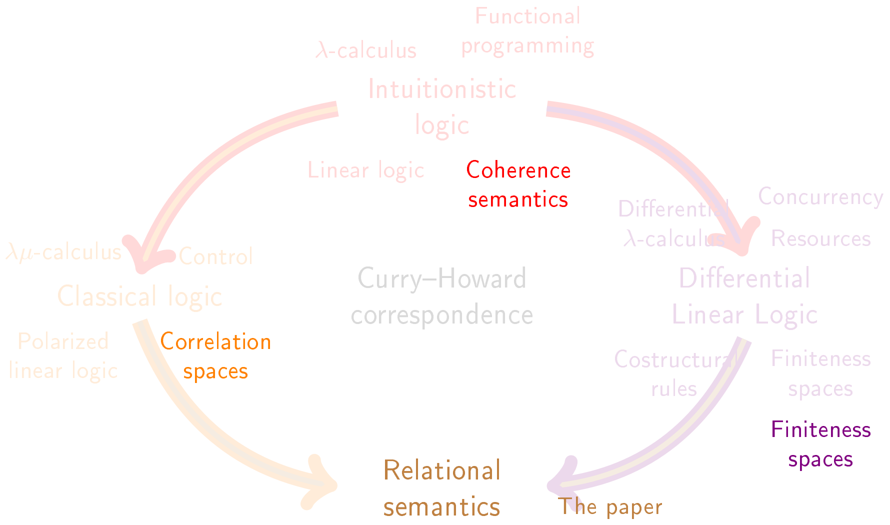
Context, Motivations



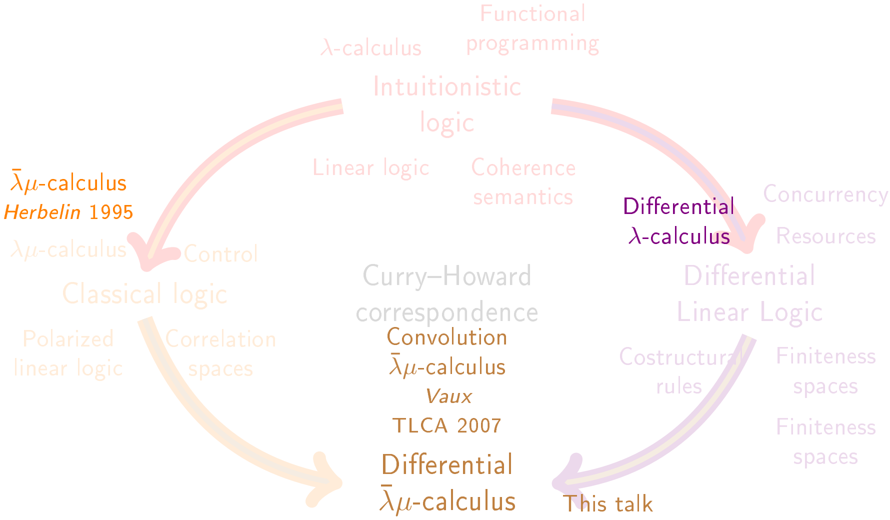
Context, Motivations, Outline



Context, Motivations, Outline



Context, Motivations, Outline

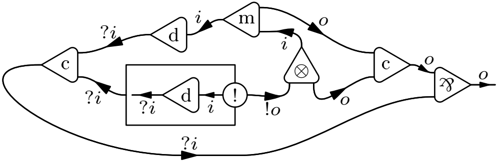


\tableofcontents

\tableofcontents

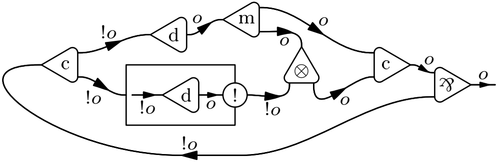
Interaction nets

An interaction net = a set of cells and wires between them (dangling wires and loops are allowed). *Lafont*, 1995



Interaction nets

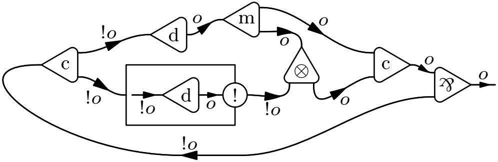
An interaction net = a set of cells and wires between them (dangling wires and loops are allowed). *Lafont*, 1995



Reversing orientation = dual typing.

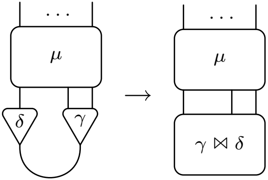
Interaction nets

An interaction net = a set of cells and wires between them (dangling wires and loops are allowed). *Lafont*, 1995



Reversing orientation = dual typing.

The core of cut elimination:



Polarization and pure types

Polarized formulas

$$\begin{array}{rcl} P, Q & ::= & X^\perp \mid P \otimes Q \mid P \oplus Q \mid \,!N \\ M, N & ::= & X \mid M \parr N \mid M \,\&\, N \mid \,?P \end{array}$$

- The polarized fragment of (ME)LL contains the pure typed λ-calculus: $M \Rightarrow N = \,!M \multimap N = \,?M^\perp \parr N$.
- It enjoys a very nice proof theory and admits structural rules on negative formulas, if not for atoms. It is the basis of Polarized Linear Logic *Laurent*, 2002.

Polarization and pure types

Polarized formulas

$$\begin{array}{rcl} P, Q & ::= & X^\perp \mid P \otimes Q \mid P \oplus Q \mid !N \\ M, N & ::= & X \mid M \parr N \mid M \,\&\, N \mid ?P \end{array}$$

Pure types (MELL for the pure untyped λ-calculus)

Let $o = ?o^\perp \parr o$ (type of all λ-terms) *Danos–Regnier*, 1990
and then $!o$ (type of arguments)
and their duals: $i = o^\perp$ (contexts) and $?i$ (free variables)

positive: i negative: o of course: $!o$ why not: $?i$

Polarization and pure types

Polarized formulas

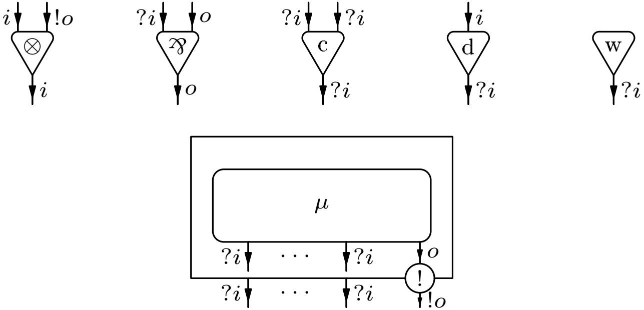
$$P, Q \quad ::= \quad X^{\perp} \mid P \otimes Q \mid P \oplus Q \mid !N$$
$$M, N \quad ::= \quad X \mid M \parr N \mid M \,\&\, N \mid ?P$$

Pure types (MELL for the pure untyped λ-calculus)

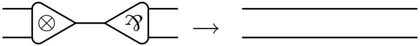
Let $o = ?o^{\perp} \parr o$ (type of all λ-terms) *Danos–Regnier*, 1990
and then $!o$ (type of arguments)
and their duals: $i = o^{\perp}$ (contexts) and $?i$ (free variables)

positive: i negative: o of course: $!o$ why not: $?i$

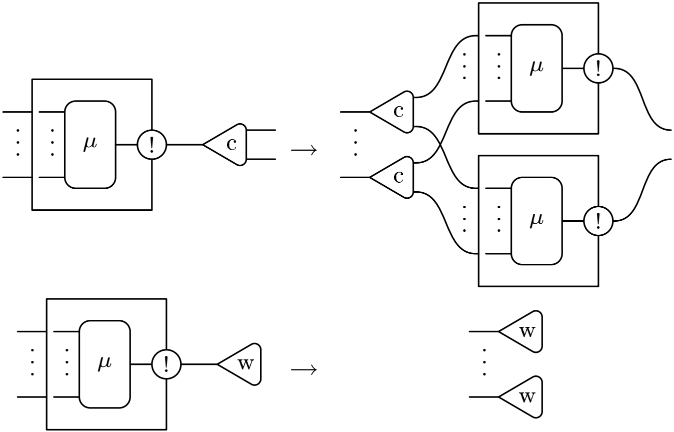
Linear logic nets



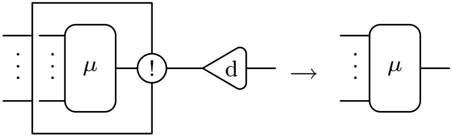
Multiplicative reduction $\langle \parr , \otimes \rangle$



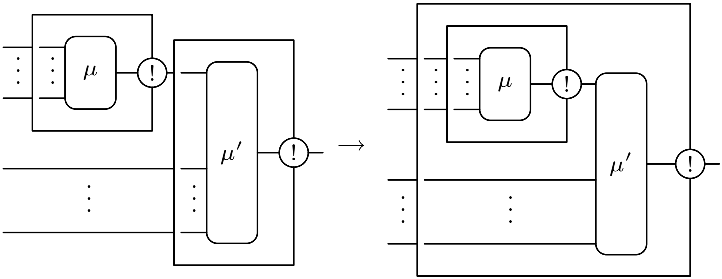
Duplication $\langle ! , \mathrm{c} \rangle$ and erasing $\langle ! , \mathrm{w} \rangle$



Box opening $\langle ! , \mathrm{d} \rangle$



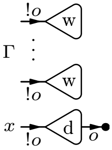
Box commutation $\langle ! , ! \rangle$



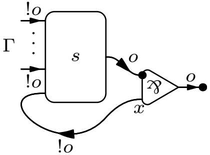
λ-calculus

Danos–Regnier, 1990

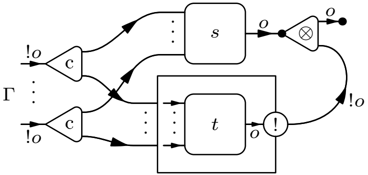
$$\frac{}{\Gamma, x \vdash x} \quad \rightsquigarrow$$



$$\frac{\Gamma, x \vdash s}{\Gamma \vdash \lambda x\, s} \quad \rightsquigarrow$$



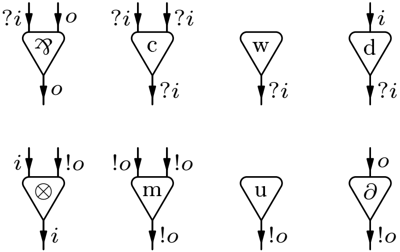
$$\frac{\Gamma \vdash s \qquad \Gamma \vdash t}{\Gamma \vdash (s)\, t} \quad \rightsquigarrow$$



\tableofcontents

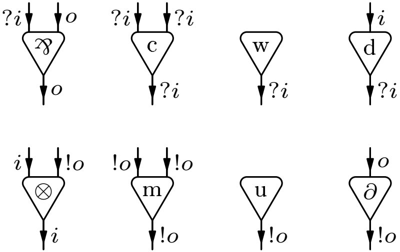
Differential nets

Ehrhard–Regnier, 2006



Differential nets

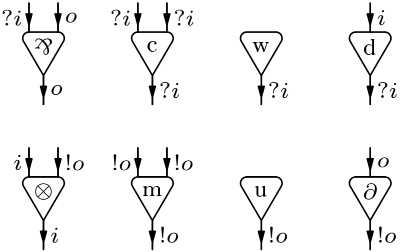
Ehrhard–Regnier, 2006



∂	:	A	$\multimap$	$!A$	derivative at 0
m	:	$!A \otimes !A$	$\multimap$	$!A$	sum of arguments of type A
u	:	$\mathbf{1}$	$\multimap$	$!A$	zero argument of type A

Differential nets

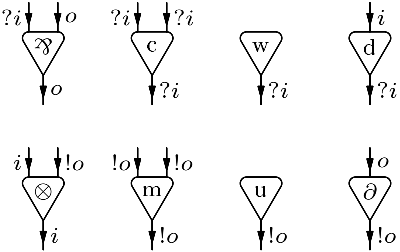
Ehrhard–Regnier, 2006



∂	:	o	$\multimap$	$!o$	make singleton packet
m	:	$!o \otimes !o$	$\multimap$	$!o$	merge
u	:	$\mathbf{1}$	$\multimap$	$!o$	empty packet

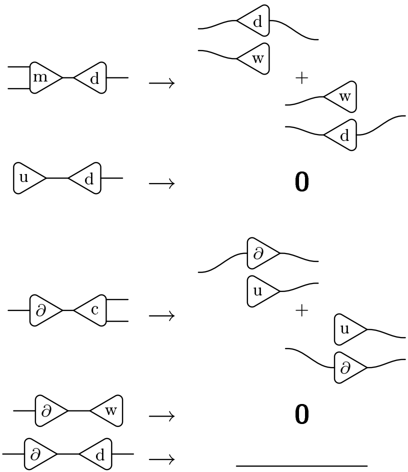
Differential nets

Ehrhard–Regnier, 2006



d	:	$!o$	$\multimap$	o	open singleton packet
c	:	$!o$	$\multimap$	$!o \otimes !o$	split
w	:	$!o$	$\multimap$	$\mathbf{1}$	erase

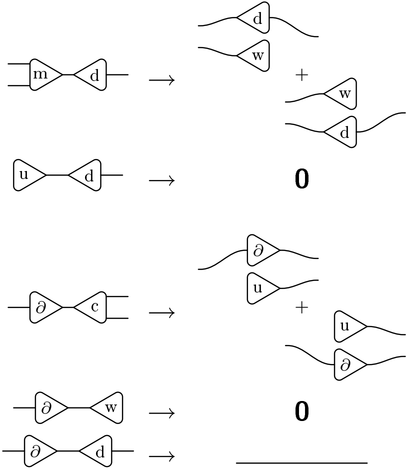
Routing and communication



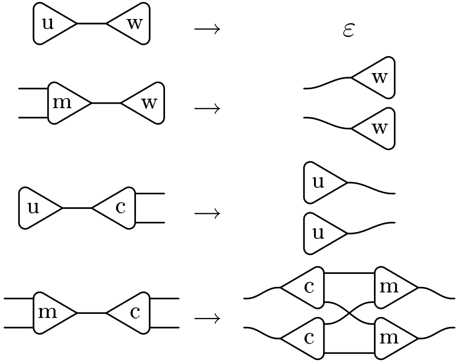
Routing and communication

Sum $\simeq$ non-deterministic choice

0 $\simeq$ failure

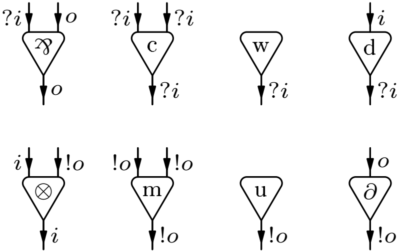


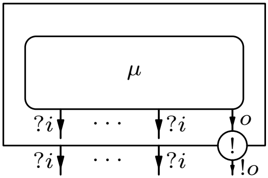
Bialgebras à la Hopf



Differential nets

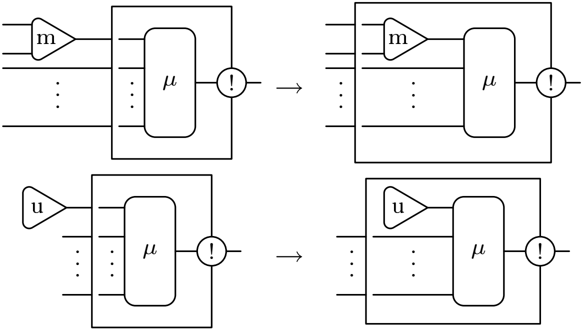
Ehrhard–Regnier, 2006



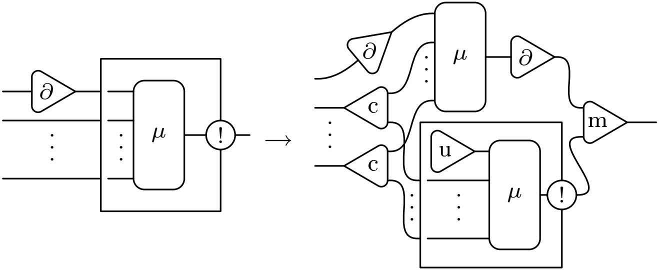


Vaux, PhD, 2007
Tranquilli, PhD, 2009
Pagani, TLCA 2009

Commutations



Chain law



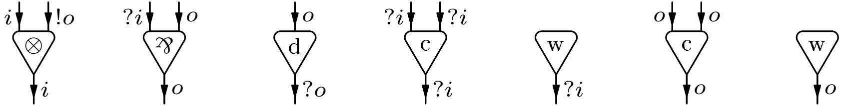
$$(f[u])' = f'[u] \times u' = (\mathrm{D}f \cdot u')[u]$$

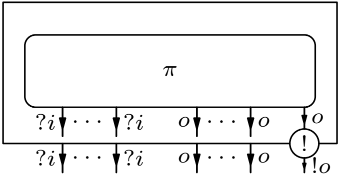
\tableofcontents

Polarized linear logic

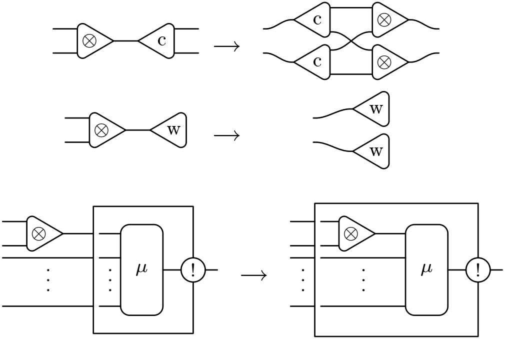
Laurent, PhD, 2002

Allow structural rules on negative types.

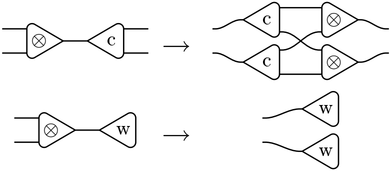




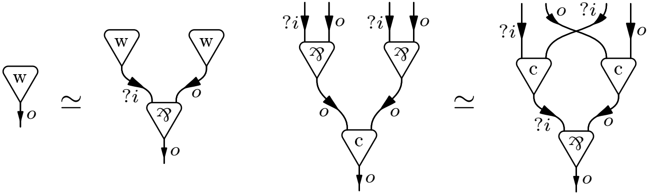
Polarized linear logic



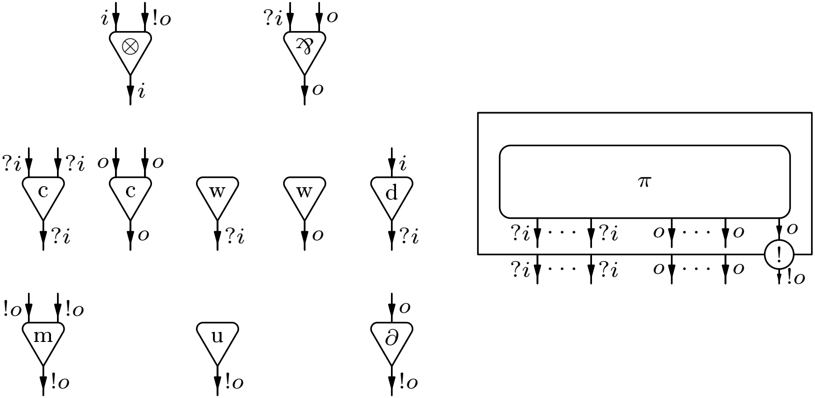
Polarized linear logic



Structural equivalence

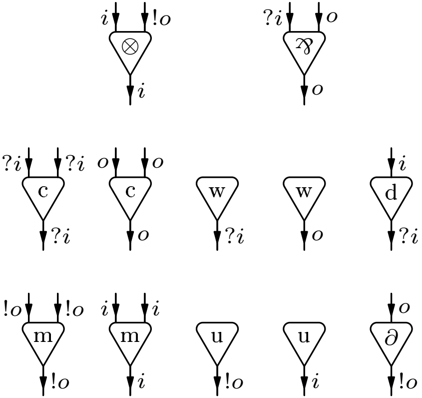


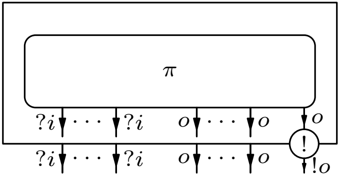
Differential nets ∪ Polarized nets



- Corresponds with the differential $\lambda\mu$-calculus.
- No new logical feature or computational effect.

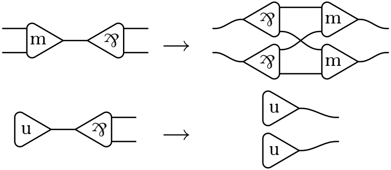
Polarized differential nets



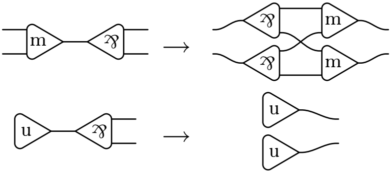


- Polarized costructural rules.

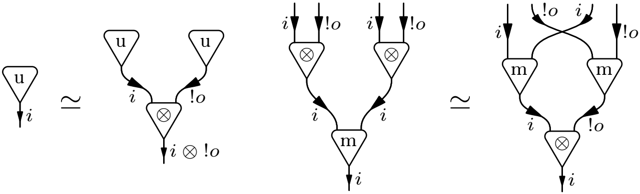
Cut elimination



Cut elimination



Structural equivalence



\tableofcontents

\tableofcontents

$\bar{\lambda}\mu$-calculus

Herbelin, PhD, 1995

$$
\begin{array}{lrcl}
\text{Terms:} & s, t & ::= & x \mid \lambda x\, s \mid \mu\alpha\, c \\
\text{Stacks:} & e & ::= & \alpha \mid s \cdot e \\
\text{Commands:} & c & ::= & \langle s \,,\, e \rangle
\end{array}
$$

$\bar{\lambda}\mu$-calculus: a term calculus for sequents *Herbelin*, PhD, 1995

Terms:	s, t	$::=$	$x \mid \lambda x\, s \mid \mu\alpha\, c$	*(conclusions)*
Stacks:	e	$::=$	$\alpha \mid s \cdot e$	*(hypotheses)*
Commands:	c	$::=$	$\langle s \,,\, e \rangle$	*(cuts)*

$\bar{\lambda}\mu$-calculus: a term calculus for sequents *Herbelin*, PhD, 1995

Terms:	s, t	$::=$	$x \mid \lambda x\, s \mid \mu\alpha\, c$	*(conclusions)*
Stacks:	e	$::=$	$\alpha \mid s \cdot e$	*(hypotheses)*
Commands:	c	$::=$	$\langle s\, ,\, e \rangle$	*(cuts)*

$$\begin{array}{rcl} \langle \lambda x\, s\, ,\, t \cdot e \rangle & \rightarrow & \langle s\,[t/x]\, ,\, e \rangle \\ \langle \mu\alpha\, c\, ,\, e \rangle & \rightarrow & c\,[e/\alpha] \end{array}$$

$\bar{\lambda}\mu$-calculus: a term calculus for sequents *Herbelin*, PhD, 1995

$$
\begin{array}{llcll}
\text{Terms:} & s, t & ::= & x \mid \lambda x\, s \mid \mu\alpha\, c & \textit{(conclusions)} \\
\text{Stacks:} & e & ::= & \alpha \mid s \cdot e & \textit{(hypotheses)} \\
\text{Commands:} & c & ::= & \langle s \,,\, e \rangle & \textit{(cuts)}
\end{array}
$$

$$
\begin{array}{rcl}
\langle \lambda x\, s \,,\, t \cdot e \rangle & \to & \langle s\,[t/x] \,,\, e \rangle \\
\langle \mu\alpha\, c \,,\, e \rangle & \to & c\,[e/\alpha]
\end{array}
$$

Encoding λ-calculus application:

$$(s)\, t = \mu\alpha\, \langle s \,,\, t \cdot \alpha \rangle$$

$\bar{\lambda}\mu$-calculus: a term calculus for sequents *Herbelin*, PhD, 1995

$$\begin{array}{llcll} \text{Terms:} & s, t & ::= & x \mid \lambda x\, s \mid \mu\alpha\, c & \textit{(conclusions)} \\ \text{Stacks:} & e & ::= & \alpha \mid s \cdot e & \textit{(hypotheses)} \\ \text{Commands:} & c & ::= & \langle s\, ,\, e \rangle & \textit{(cuts)} \end{array}$$

$$\begin{array}{rcl} \langle \lambda x\, s\, ,\, t \cdot e \rangle & \rightarrow & \langle s\, [t/x]\, ,\, e \rangle \\ \langle \mu\alpha\, c\, ,\, e \rangle & \rightarrow & c\, [e/\alpha] \end{array}$$

Encoding λ-calculus application:

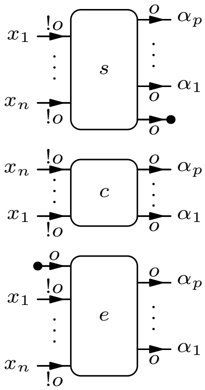
$$(s)\, t = \mu\alpha\, \langle s\, ,\, t \cdot \alpha \rangle$$

η-expansion:

$$s \leftarrow_\eta \lambda x\, \mu\alpha\, \langle s\, ,\, x \cdot \alpha \rangle$$

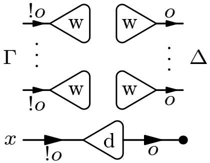
Translation into in polarized nets

Simplified version of Laurent's translation of Curien–Herbelin's $\bar{\lambda}\mu$-calculus. *Laurent*, PhD, 2002

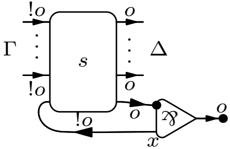


Translation: terms

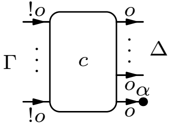
$\Gamma \vdash x \mid \Delta \quad \rightsquigarrow$



$\Gamma \vdash \lambda x\, s \mid \Delta \quad \rightsquigarrow$

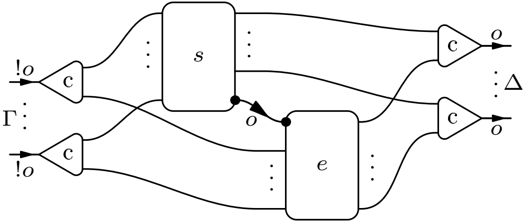


$\Gamma \vdash \mu\alpha\, c \mid \Delta \quad \rightsquigarrow$



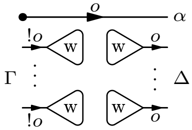
Translation: commands

$\langle s \, , e \rangle : (\Gamma \vdash \Delta) \quad \rightsquigarrow$

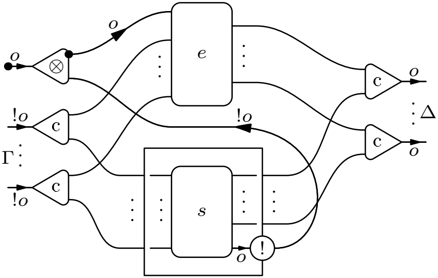


Translation: stacks

$\Gamma \mid \alpha \vdash \Delta \quad \rightsquigarrow$



$\Gamma \mid s \cdot e \vdash \Delta \quad \rightsquigarrow$



\tableofcontents

Convolution $\bar{\lambda}\mu$-calculus

Vaux, TLCA 2007

Computational interpretation of polarized costructural rules

Positive formulas (i.e. i) appear as the type of stacks.

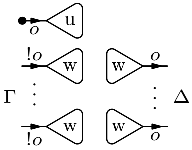
Convolution $\bar{\lambda}\mu$-calculus

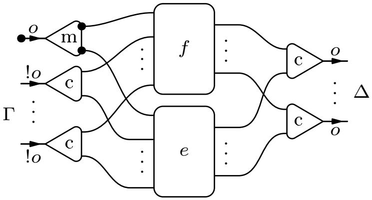
Vaux, TLCA 2007

Computational interpretation of polarized costructural rules

Positive formulas (i.e. i) appear as the type of stacks.

$$\begin{array}{rcl} s,t & ::= & x \mid \lambda x\, s \mid \mu\alpha\, c \\ e,f & ::= & \alpha \mid s\cdot e \mid \mathbf{1} \mid e * f \\ c & ::= & \langle s\, ,\, e\rangle \end{array}$$

$\mathbf{1} \rightsquigarrow$ 

$e * f \rightsquigarrow$ 

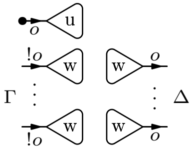
Convolution $\bar{\lambda}\mu$-calculus

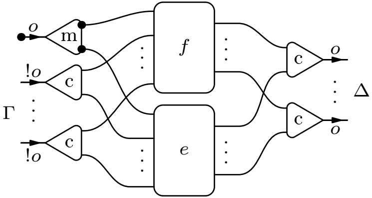
Vaux, TLCA 2007

Computational interpretation of polarized costructural rules

Positive formulas (i.e. i) appear as the type of stacks.

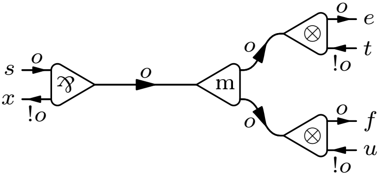
$$
\begin{aligned}
s,t &::= x \mid \lambda x\, s \mid \mu\alpha\, c \mid \mathbf{0} \mid s+t\\
e,f &::= \alpha \mid s\cdot e \mid \mathbf{1} \mid e*f \mid \mathbf{0} \mid e+f\\
c &::= \langle s\,,\,e\rangle \mid \mathbf{0} \mid c+c'
\end{aligned}
$$

$\mathbf{1} \rightsquigarrow$ 

$e*f \rightsquigarrow$ 

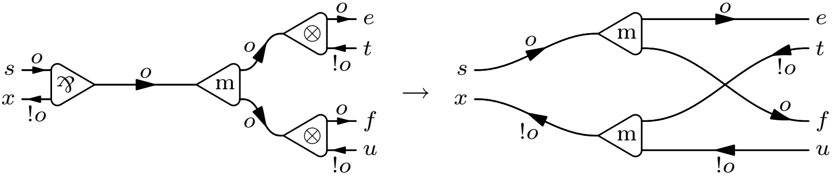
Routing $\rightsquigarrow$ sums.

Reduction



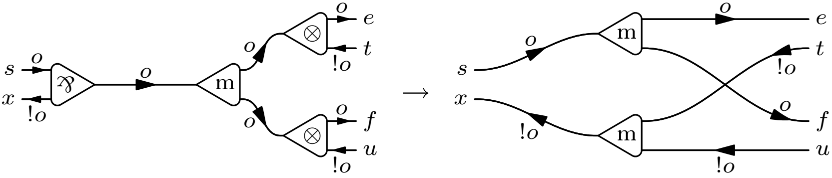
$$\langle \lambda x\, s\,,\, (t \cdot e) * (u \cdot f) \rangle$$

Reduction



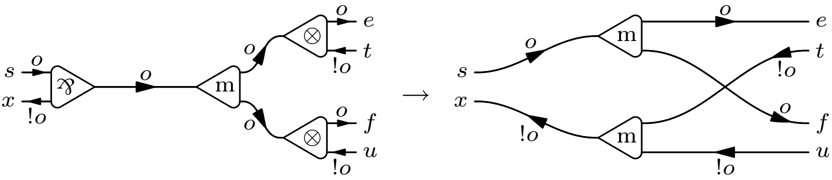
$$\langle \lambda x\, s\,,\, (t \cdot e) * (u \cdot f) \rangle$$

Reduction



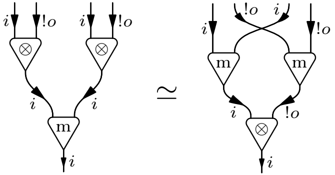
$$\langle \lambda x\, s\, ,\, (t \cdot e) * (u \cdot f) \rangle \rightarrow \langle s\,[t + u/x]\, ,\, e * f \rangle$$

Reduction



$$\langle \lambda x\, s \,,\, (t \cdot e) * (u \cdot f) \rangle \;\rightarrow\; \langle s\,[t + u/x] \,,\, e * f \rangle$$

$(u{\cdot}e){*}(v{\cdot}f) \simeq (t{+}u){\cdot}(e{*}f)$ cf.



Reduction

$$
\begin{aligned}
\langle \mu\alpha\, c \,,\, e \rangle &\rightarrow c\,[e/\alpha] \\
\langle \lambda x\, s \,,\, (t_1 \cdot e_1) * \cdots * (t_k \cdot e_k) \rangle &\rightarrow \langle s\,[t_1 + \cdots + t_k / x] \,,\, e_1 * \cdots * e_k \rangle
\end{aligned}
$$

Reduction

$$
\begin{aligned}
\langle \mu\alpha\, c \,,\, e\rangle &\rightarrow c\,[e/\alpha] \\
\langle \lambda x\, s \,,\, (t \cdot e) * f\rangle &\rightarrow \langle \lambda y\, \mu\beta\, \langle s\,[y + t/x] \,,\, e * \beta\rangle \,,\, f\rangle \\
\langle \lambda x\, s \,,\, \mathbf{1}\rangle &\rightarrow \langle s\,[\mathbf{0}/x] \,,\, \mathbf{1}\rangle
\end{aligned}
$$

Reduction

$$\begin{array}{rcl} \langle \mu\alpha\, c\,,\, e\rangle & \rightarrow & c\,[e/\alpha] \\ \langle \lambda x\, s\,,\, (t \cdot e) * f\rangle & \rightarrow & \langle \lambda y\, \mu\beta\, \langle s\,[y + t/x]\,,\, e * \beta\rangle\,,\, f\rangle \\ \langle \lambda x\, s\,,\, \mathbf{1}\rangle & \rightarrow & \langle s\,[\mathbf{0}/x]\,,\, \mathbf{1}\rangle \end{array}$$

The same, modulo:

$$\langle s\,,\, e * f\rangle \leftarrow_{\eta'} \langle \lambda x\, \mu\alpha\, \langle s\,,\, e * (x \cdot \alpha)\rangle\,,\, f\rangle\,.$$

\tableofcontents

De-synchronisation

Recall that: $(u \cdot e) * (v \cdot f) \simeq (t + u) \cdot (e * f)$

De-synchronisation

Recall that: $(u \cdot e) * (v \cdot f) \simeq (t + u) \cdot (e * f)$

Let:

$$\begin{aligned} s^! &= s \cdot \mathbf{1} \\ \uparrow e &= \mathbf{0} \cdot e \end{aligned}$$

De-synchronisation

Recall that: $(u \cdot e) * (v \cdot f) \simeq (t + u) \cdot (e * f)$

Let:

$$\begin{aligned} s^! &= s \cdot \mathbf{1} \\ \uparrow e &= \mathbf{0} \cdot e \end{aligned}$$

Then: $s \cdot e \simeq s^! * \uparrow e$

De-synchronisation

Recall that: $(u \cdot e) * (v \cdot f) \simeq (t + u) \cdot (e * f)$

Let:

$$\begin{aligned} s^! &= s \cdot \mathbf{1} \\ \uparrow e &= \mathbf{0} \cdot e \end{aligned}$$

Then: $s \cdot e \simeq s^! * \uparrow e$

$$\begin{aligned} \left\langle \lambda x\, s \,,\, t^! * f \right\rangle &\to \langle \lambda x\, s\,[x + t/x] \,,\, f \rangle \\ \langle \lambda x\, s \,,\, \uparrow e * f \rangle &\to \langle \lambda x\, \mu\alpha\, \langle s \,,\, e * \alpha \rangle \,,\, f \rangle\,. \end{aligned}$$

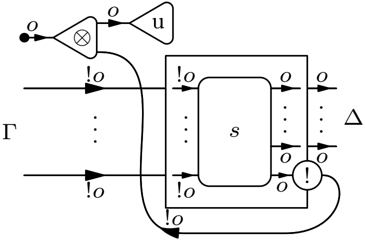
De-synchronisation

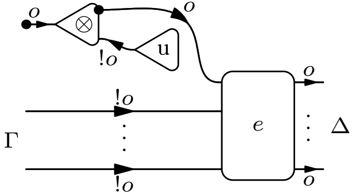
Recall that: $(u \cdot e) * (v \cdot f) \simeq (t + u) \cdot (e * f)$

Let:

$$\begin{aligned} s^{!} &= s \cdot \mathbf{1} \\ \uparrow e &= \mathbf{0} \cdot e \end{aligned}$$

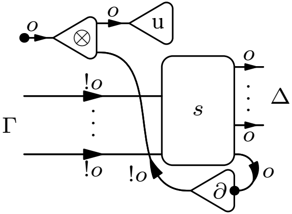
Then: $s \cdot e \simeq s^{!} * \uparrow e$





Differential $\bar{\lambda}\mu$-calculus

Linear argument for derivative: $[s]$



Reduction

$$
\begin{array}{rcl}
\langle \lambda x\, s\,,\, \mathbf{1} \rangle & \rightarrow & \langle s\,[\mathbf{0}/x]\,,\, \mathbf{1} \rangle \\
\left\langle \lambda x\, s\,,\, t^{!} * f \right\rangle & \rightarrow & \langle \lambda x\, s\,[t + x/x]\,,\, f \rangle \\
\langle \lambda x\, s\,,\, \uparrow e * f \rangle & \rightarrow & \langle \lambda x\, \mu\alpha\, \langle s\,,\, e * \alpha \rangle\,,\, f \rangle \\
\langle \lambda x\, s\,,\, [t] * f \rangle & \rightarrow & \left\langle \lambda x \left(\dfrac{\partial\, s}{\partial\, x} \cdot t \right),\, f \right\rangle \\
\langle \mu\alpha\, c\,,\, e \rangle & \rightarrow & \langle c\,,\, e \rangle_{\alpha}
\end{array}
$$

Linear substitution *aka.* partial derivative

Cf. differential λ-calculus *Ehrhard–Regnier*, 2003

$$\frac{\partial\, y}{\partial\, x}\cdot t = \delta_{x,y} t \qquad \frac{\partial\, \alpha}{\partial\, x}\cdot t = \mathbf{0}$$

$$\frac{\partial\, \lambda y\, s}{\partial\, x}\cdot t = \lambda y\left(\frac{\partial\, s}{\partial\, x}\cdot t\right) \qquad \frac{\partial\,[s]}{\partial\, x}\cdot t = \left[\frac{\partial\, s}{\partial\, x}\cdot t\right]$$

$$\frac{\partial\, \mu\alpha\, c}{\partial\, x}\cdot t = \mu\alpha\left(\frac{\partial\, c}{\partial\, x}\cdot t\right) \qquad \frac{\partial\, s^!}{\partial\, x}\cdot t = s^! * \left[\frac{\partial\, S}{\partial\, x}\cdot t\right]$$

$$\frac{\partial\, \langle s\,,\, e\rangle}{\partial\, x}\cdot t = \left\langle \frac{\partial\, s}{\partial\, x}\cdot t\,,\, e\right\rangle + \left\langle s\,,\, \frac{\partial\, e}{\partial\, x}\cdot t\right\rangle \qquad \frac{\partial\, \uparrow e}{\partial\, x}\cdot t = \uparrow\left(\frac{\partial\, e}{\partial\, x}\cdot t\right)$$

$$\frac{\partial\, \mathbf{0}}{\partial\, x}\cdot t = \mathbf{0} \qquad \frac{\partial\, \mathbf{1}}{\partial\, x}\cdot t = \mathbf{0}$$

$$\frac{\partial\, \theta+\theta'}{\partial\, x}\cdot t = \frac{\partial\, \theta}{\partial\, x}\cdot t + \frac{\partial\, \theta'}{\partial\, x}\cdot t \qquad \frac{\partial(e * f)}{\partial\, x}\cdot t = \left(\frac{\partial\, e}{\partial\, x}\cdot t\right) * f + e * \left(\frac{\partial\, f}{\partial\, x}\cdot t\right)$$

Linear substitution *aka.* partial derivative

Cf. differential λ-calculus *Ehrhard–Regnier*, 2003

$$\frac{\partial\, y}{\partial\, x}\cdot t = \delta_{x,y}\, t \qquad \frac{\partial\, \alpha}{\partial\, x}\cdot t = \mathbf{0}$$

$$\frac{\partial\, \lambda y\, s}{\partial\, x}\cdot t = \lambda y\left(\frac{\partial\, s}{\partial\, x}\cdot t\right) \qquad \frac{\partial\, [s]}{\partial\, x}\cdot t = \left[\frac{\partial\, s}{\partial\, x}\cdot t\right]$$

$$\frac{\partial\, \mu\alpha\, c}{\partial\, x}\cdot t = \mu\alpha\left(\frac{\partial\, c}{\partial\, x}\cdot t\right) \qquad \frac{\partial\, s^!}{\partial\, x}\cdot t = s^! * \left[\frac{\partial\, S}{\partial\, x}\cdot t\right]$$

$$\frac{\partial\, \langle s\,,\, e\rangle}{\partial\, x}\cdot t = \left\langle \frac{\partial\, s}{\partial\, x}\cdot t\,,\, e\right\rangle + \left\langle s\,,\, \frac{\partial\, e}{\partial\, x}\cdot t\right\rangle \qquad \frac{\partial\, {\uparrow} e}{\partial\, x}\cdot t = {\uparrow}\left(\frac{\partial\, e}{\partial\, x}\cdot t\right)$$

$$\frac{\partial\, \mathbf{0}}{\partial\, x}\cdot t = \mathbf{0} \qquad \frac{\partial\, \mathbf{1}}{\partial\, x}\cdot t = \mathbf{0}$$

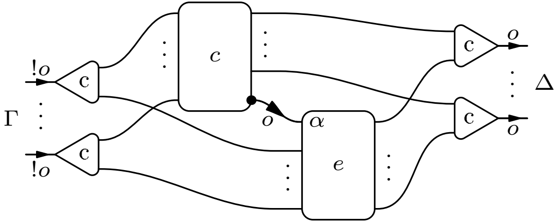
$$\frac{\partial\, \theta + \theta'}{\partial\, x}\cdot t = \frac{\partial\, \theta}{\partial\, x}\cdot t + \frac{\partial\, \theta'}{\partial\, x}\cdot t \qquad \frac{\partial (e * f)}{\partial\, x}\cdot t = \left(\frac{\partial\, e}{\partial\, x}\cdot t\right) * f + e * \left(\frac{\partial\, f}{\partial\, x}\cdot t\right)$$

Reduction

$$
\begin{array}{rcl}
\langle \lambda x\, s\,,\, \mathbf{1} \rangle & \rightarrow & \langle s\, [\mathbf{0}/x]\,,\, \mathbf{1} \rangle \\
\left\langle \lambda x\, s\,,\, t^{!} * f \right\rangle & \rightarrow & \langle \lambda x\, s\, [t + x/x]\,,\, f \rangle \\
\langle \lambda x\, s\,,\, \uparrow e * f \rangle & \rightarrow & \langle \lambda x\, \mu\alpha\, \langle s\,,\, e * \alpha \rangle\,,\, f \rangle \\
\langle \lambda x\, s\,,\, [t] * f \rangle & \rightarrow & \left\langle \lambda x \left(\frac{\partial\, s}{\partial\, x} \cdot t \right),\, f \right\rangle \\
\langle \mu\alpha\, c\,,\, e \rangle & \rightarrow & \langle c\,,\, e \rangle_{\alpha}
\end{array}
$$

Named cut

$\langle c \,,\, e \rangle_\alpha \rightsquigarrow$



Reduction

$$
\begin{array}{rcl}
\langle \lambda x\, s\, ,\, \mathbf{1} \rangle & \to & \langle s\, [\mathbf{0}/x]\, ,\, \mathbf{1} \rangle \\
\left\langle \lambda x\, s\, ,\, t^{!} * f \right\rangle & \to & \langle \lambda x\, s\, [t + x/x]\, ,\, f \rangle \\
\langle \lambda x\, s\, ,\, \uparrow e * f \rangle & \to & \langle \lambda x\, \mu\alpha\, \langle s\, ,\, e * \alpha \rangle\, ,\, f \rangle \\
\langle \lambda x\, s\, ,\, [t] * f \rangle & \to & \left\langle \lambda x \left(\dfrac{\partial\, s}{\partial\, x} \cdot t \right) ,\, f \right\rangle \\
\langle \mu\alpha\, c\, ,\, \beta \rangle & \to & c\, [\beta/\alpha] \\
\left\langle \mu\alpha\, c\, ,\, s^{!} \right\rangle & \to & c\, \left[s^{!}/\alpha \right] \\
\langle \mu\alpha\, c\, ,\, \mathbf{1} \rangle & \to & c\, [\mathbf{1}/\alpha] \\
\langle \mu\alpha\, c\, ,\, \uparrow e \rangle & \to & \langle \mu\alpha\, c\, [\uparrow\alpha\, /\alpha]\, ,\, e \rangle \\
\langle \mu\alpha\, c\, ,\, (e * f) \rangle & \to & \langle \mu\alpha''\, \langle \mu\alpha'\, c\, [\alpha' * \alpha''/\alpha]\, ,\, e \rangle\, ,\, f \rangle \\
\langle \mu\alpha\, c\, ,\, [s] \rangle & \to & \left(\dfrac{\partial\, c}{\partial\, \alpha} \cdot [s] \right) [\mathbf{1}/\alpha]
\end{array}
$$

\tableofcontents

Outcome

Polarized differential logic

- Proper extension of both differential and polarized linear logics
- Preserves the symmetries of differential linear logic
- Breaks the intuitionnistic flavour of polarization (cf. paper)
- Non uniform semantics in the relational model (cf. paper)

Differential $\bar{\lambda}\mu$-calculus

- Polarized costructural rules = [Vaux, TLCA 2007]
- Stacks as “bags of resources”: breaks sequentiality between arguments
- *A good playground to explore concurrency in a functional, logically founded setting.*

Towards concurrency

Resource λ-calculus

Boudol, CONCUR 1993

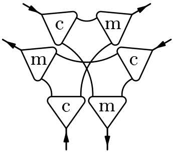
$$\begin{array}{rcl} s & ::= & x \mid \lambda x\, s \mid (s)\, S \\ S & ::= & \mathbf{1} \mid S * S \mid [s] \mid s^! \end{array}$$

As discriminative as Milner's π-calculus w.r.t. λ-calculus
Boudol–Laneve, 1995

Translation: $(s)\, S \rightsquigarrow \mu\alpha\, \langle s\,,\, S \cdot \alpha \rangle$

DiLL as a model of concurrency

Ehrhard–Laurent, CONCUR 2007



- Costructural rules are crucial.
- Also makes use of *non sequential* nets. How much is that essential ?
- What is the concurrent expressivity of the diff. $\bar{\lambda}\mu$-calculus (fragment of π-calculus) ?
- What about concurrent semantics (event structures, cf. recent work by Mazza) ?

A word on the Taylor expansion

The finitary fragment of the resource λ-calculus encodes the pure λ-calculus *Ehrhard–Regnier*, 2006

Similarly

$$s^{!} \simeq \sum_{n=0}^{\infty} \frac{1}{n!} [s]^n$$

To be related with Carvalho's study of LL promotion in models of DiLL... *de Carvalho*, PhD, 2007

...and recent work by Tasson and Pagani demonstrating an inverse Taylor expansion in DiLL *Tasson–Pagani*, LICS, 2009

Can this bring new light on the properties of replication in process algebras?

Thank you

for your attention

e grazie a Michele

L. Vaux
lionel.vaux@univ-savoie.fr