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

Lionel Vaux

Institut de Mathématiques de Luminy, Marseille
vaux@iml.univ-mrs.fr

Typed Lambda Calculi and Applications, 2007, Paris

\tableofcontents

$\bar\lambda\mu$-calculus [Her95]

Morphology :

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

Reductions :

$$\begin{array}{rcll} \langle \lambda x\, s\,,\, t \cdot e\rangle & \to & \langle s\,[x := t]\,,\, e\rangle & (\beta) \\ \langle \mu\alpha\, c\,,\, e\rangle & \to & c\,[\alpha := e] & (\mu) \quad . \end{array}$$

Hugo Herbelin.
Séquents qu'on calcule.
PhD Thesis, Université Paris 7, 1995.

Encoding ordinary β-reduction

- We can encode : $(s)\, t = \mu\alpha \langle s \,,\, t \cdot \alpha \rangle$.
- Then

$$(\lambda x\, s)\, t = \mu\alpha \langle \lambda x\, s \,,\, t \cdot \alpha \rangle \to \mu\alpha \langle s\, [x := t] \,,\, \alpha \rangle \quad .$$

- η-reduction on names :

$$\mu\alpha \langle u \,,\, \alpha \rangle \to_\eta u$$

if α is not free in u.

η-expansion

- If α is not free in s :

  $$s \leftarrow_\eta \mu\alpha \langle s \,,\, \alpha \rangle \quad .$$

  If moreover x is not free in s :

  $$s \leftarrow_\eta \lambda x \, \mu\alpha \langle s \,,\, x \cdot \alpha \rangle = \lambda x \, (s) \, x \quad .$$

- Notice that :

  $$\langle \mu\alpha \langle s \,,\, \alpha \rangle \,,\, e \rangle \to \langle s \,,\, e \rangle$$

  and

  $$\langle \lambda x \, \mu\alpha \langle s \,,\, x \cdot \alpha \rangle \,,\, t \cdot e \rangle \to \langle \mu\alpha \langle s \,,\, t \cdot \alpha \rangle \,,\, e \rangle \to \langle s \,,\, t \cdot e \rangle \quad .$$

Monoid of terms

In differential λ-calculus [ER03], sums of terms arise naturally.

- Algebraic interpretation :
  functions with values in a monoid form a monoid
  with $(f + g)[x] = f[x] + g[x]$.
- In λ-calculus :

$$\lambda x\,(s + t) = \lambda x\, s + \lambda x\, t \text{ and } (s + t)\, u = (s)\, u + (t)\, u \quad .$$

- Another interpretation : nondeterministic choice.

Thomas Ehrhard and Laurent Regnier.
The differential lambda-calculus.
TCS, 309 :1–41, 2003.

Monoid of terms

In differential λ-calculus [ER03], sums of terms arise naturally.

- Algebraic interpretation :
  functions with values in a monoid form a monoid
  with $(f+g)[x] = f[x] + g[x]$.
- In λ-calculus :

$$\lambda x\,(s+t) = \lambda x\,s + \lambda x\,t \text{ and } (s+t)\,u = (s)\,u + (t)\,u \quad .$$

- Another interpretation : nondeterministic choice.
- One can extend this to linear combinations
  (your mileage may vary [Vau07]).

Lionel Vaux.
On linear combinations of λ-terms.
RTA 2007.

Convolution product on boxes

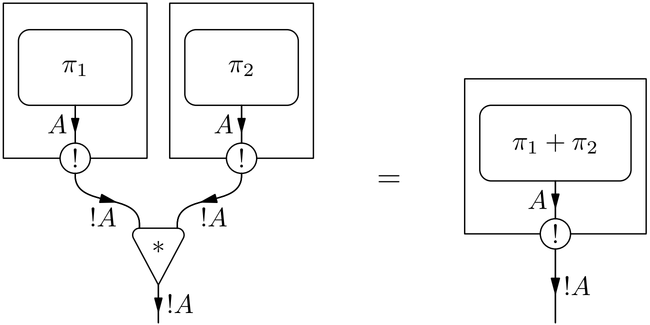
- In models of differential λ-calculus, exponential types ($!A$) are endowed with a monoidal structure.
- Much like the convolution product of distributions [Ehr01, Ehr05].

Thomas Ehrhard.
On Köthe sequence spaces and linear logic.
MSCS, 2001.

Thomas Ehrhard.
Finiteness spaces.
MSCS, 2005.

Convolution product on boxes

- In models of differential λ-calculus, exponential types (!*A*) are endowed with a monoidal structure.
- Much like the convolution product of distributions [Ehr01, Ehr05].
- Main intuition : a product of boxes is a sum inside a box.



Back to a term calculus

- Boxes in LL $\simeq$ arguments of applications in λ-calculus.
- This feature is used in differential λ-calculus, but in a hidden way.
- λ-calculus does not provide easy access to arguments ($\neq$ subterms).
- In $\bar{\lambda}(\mu)$-calculus, *lists* of arguments are explicitly part of the syntax.

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

Introduce a monoid operation $(*, \mathbf{1})$ on contexts such that :

- the set of contexts with $(+, \mathbf{0}, *, \mathbf{1})$ is a semiring with zero and unit (aka. *rig*) ;
- $(s \cdot e) * (t \cdot f)$ behaves like $(s + t) \cdot (e * f)$.
- Corresponds to an extension of differential interaction nets, along the lines of Laurent's polarization of LL proof nets.

Thomas Ehrhard and Laurent Regnier.
Differential interaction nets.
ENTCS, 2005.

Olivier Laurent.
Polarized proof-nets and $\lambda\mu$-calculus.
TCS, 2003.

Morphology

Definition

Define terms $(s, t, u, \ldots)$, contexts $(e, f, \ldots)$ and commands $(c, d, \ldots)$:

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

Morphology

Definition

Define terms $(s, t, u, \dots)$, contexts $(e, f, \dots)$ and commands $(c, d, \dots)$:

$$
\begin{array}{rcl}
s, t & ::= & x \mid \lambda x\, s \mid \mu\alpha\, c \mid \mathbf{0} \mid s + t \\
e, f & ::= & \alpha \mid s \cdot e \mid \mathbf{1} \mid e * f \mid \mathbf{0} \mid e + f \\
c, d & ::= & \langle s \,,\, e \rangle \mid \mathbf{0} \mid c + d
\end{array}
$$

Morphology

Definition

Define terms $(s, t, u, \ldots)$, contexts $(e, f, \ldots)$ and commands $(c, d, \ldots)$:

$$\begin{array}{rcl} s, t & ::= & x \mid \lambda x\, s \mid \mu\alpha\, c \mid \mathbf{0} \mid s + t \\ e, f & ::= & \alpha \mid s \cdot e \mid \mathbf{1} \mid e * f \mid \mathbf{0} \mid e + f \\ c, d & ::= & \langle s \,,\, e \rangle \mid \mathbf{0} \mid c + d \end{array}$$

We consider terms up to usual α-equivalence, monoid structure, and

$$\begin{array}{rclrcl} \lambda x\, \mathbf{0} & = & \mathbf{0} & \lambda x\, (s + t) & = & \lambda x\, s + \lambda x\, t \\ \langle \mathbf{0} \,,\, e \rangle & = & \mathbf{0} & \langle s + t \,,\, e \rangle & = & \langle s \,,\, e \rangle + \langle t \,,\, e \rangle \\ s \cdot \mathbf{0} & = & \mathbf{0} & s \cdot (c + d) & = & s \cdot c + s \cdot d \qquad \ldots \end{array}$$

Morphology

Definition

Define terms $(s, t, u, \ldots)$, contexts $(e, f, \ldots)$ and commands $(c, d, \ldots)$:

$$\begin{array}{rcl} s, t & ::= & x \mid \lambda x\, s \mid \mu\alpha\, c \mid \mathbf{0} \mid s + t \\ e, f & ::= & \alpha \mid s \cdot e \mid \mathbf{1} \mid e * f \mid \mathbf{0} \mid e + f \\ c, d & ::= & \langle s \,,\, e \rangle \mid \mathbf{0} \mid c + d \end{array}$$

We consider terms up to usual α-equivalence, monoid structure, and

$$\begin{array}{rclrcl} \lambda x\, \mathbf{0} & = & \mathbf{0} & \lambda x\,(s + t) & = & \lambda x\, s + \lambda x\, t \\ \langle \mathbf{0} \,,\, e \rangle & = & \mathbf{0} & \langle s + t \,,\, e \rangle & = & \langle s \,,\, e \rangle + \langle t \,,\, e \rangle \\ s \cdot \mathbf{0} & = & \mathbf{0} & s \cdot (c + d) & = & s \cdot c + s \cdot d \qquad \ldots \end{array}$$

The only non linear position is that of arguments : $s \cdot e$
(cf. linear logic boxes).

About reduction rules

- Recall that in $\bar{\lambda}\mu$-calculus :

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

- Analogy :

$$(s\cdot e) * (t\cdot f) \simeq (s+t)\cdot(e*f)$$

- Hence we set :

$$\langle \lambda x\, s\,,\, (t\cdot e)*(u\cdot f)\rangle \rightsquigarrow \langle s\,[x := t+u]\,,\, e*f\rangle$$

About reduction rules

- Recall that in $\bar{\lambda}\mu$-calculus :

$$\begin{aligned} \langle \lambda x\, s \,,\, t \cdot e \rangle &\rightarrow \langle s\,[x := t] \,,\, e \rangle \\ \langle \mu\alpha\, c \,,\, e \rangle &\rightarrow c\,[\alpha := e] \end{aligned}$$

- Analogy :

$$(s \cdot e) * (t \cdot f) \simeq (s + t) \cdot (e * f)$$

- Hence we set :

$$\begin{aligned} \langle \lambda x\, s \,,\, (t \cdot e) * (u \cdot f) \rangle &\rightsquigarrow \langle s\,[x := t + u] \,,\, e * f \rangle \\ \langle \lambda x\, s \,,\, (t \cdot e) * f \rangle &\rightsquigarrow \langle \lambda y\, \mu\alpha\, \langle s\,[x := y + t] \,,\, \alpha * e \rangle \,,\, f \rangle \end{aligned}$$

Reduction

Definition

Reduction in convolution $\bar{\lambda}\mu$-calculus is given by :

$$
\begin{array}{rcll}
\langle \mu\alpha\, c \,,\, e\rangle & \to & c\,[\alpha := e] & (\mu) \\
\langle \lambda x\, s \,,\, (t \cdot e) * f\rangle & \to & \langle \lambda y\, \mu\alpha\, \langle s\,[x := y + t] \,,\, \alpha * e\rangle \,,\, f\rangle & (\beta_*) \\
\langle \lambda x\, s \,,\, \mathbf{1}\rangle & \to & \langle s\,[x := \mathbf{0}] \,,\, \mathbf{1}\rangle & (\beta_{\mathbf{1}})
\end{array}
$$

$$
\begin{array}{rcl}
\langle \lambda x\, s \,,\, t \cdot e\rangle & = & \langle \lambda x\, s \,,\, (t \cdot e) * \mathbf{1}\rangle \\
& \to & \langle \lambda x\, \mu\alpha\, \langle s\,[x := x + t] \,,\, \alpha * e\rangle \,,\, \mathbf{1}\rangle \\
& \to & \langle \mu\alpha\, \langle s\,[x := \mathbf{0} + t] \,,\, \alpha * e\rangle \,,\, \mathbf{1}\rangle \\
& \to & \langle s\,[x := \mathbf{0} + t] \,,\, \mathbf{1} * e\rangle \\
& = & \langle s\,[x := t] \,,\, e\rangle \quad .
\end{array}
$$

Reduction

Definition

Reduction in convolution $\bar{\lambda}\mu$-calculus is given by :

$$\begin{array}{rcll} \langle \mu\alpha\, c \,,\, e\rangle & \to & c\,[\alpha := e] & (\mu) \\ \langle \lambda x\, s \,,\, (t \cdot e) * f\rangle & \to & \langle \lambda y\, \mu\alpha \,\langle s\,[x := y + t] \,,\, \alpha * e\rangle \,,\, f\rangle & (\beta_*) \\ \langle \lambda x\, s \,,\, \mathbf{1}\rangle & \to & \langle s\,[x := \mathbf{0}] \,,\, \mathbf{1}\rangle & (\beta_\mathbf{1}) \end{array}$$

Theorem

Reduction is confluent.

Convolution product on distributions

- Let $\mathcal{D}$ be the vector space of infinitely differentiable functions ($\mathbf{R} \longrightarrow \mathbf{R}$) with compact support.
- Distributions are linear and continuous functionals from $\mathcal{D}$ to $\mathbf{R}$.
- Write $\langle \varphi \, , f \rangle$ for $f(\varphi)$.
- Under conditions ensuring well-definedness, one sets :

$$\langle \lambda z \, \varphi(z) \, , f * g \rangle = \langle \lambda y \, \langle \lambda x \, \varphi(x + y) \, , f \rangle \, , g \rangle \quad .$$

Laurent Schwartz.
Théorie des distributions.
Hermann, 1966.

Convolution product on contexts (1)

Commands

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

and

$$\langle \lambda y\, \mu\beta \langle \lambda x\, \mu\alpha \langle s\,[z := x + y]\, ,\, \alpha * \beta \rangle\, ,\, t \cdot e \rangle\, ,\, u \cdot f \rangle$$

both reduce to

$$\langle s\,[z := t + u]\, ,\, e * f \rangle \quad .$$

Some informal statements

On types and syntactic categories

- Terms are analogous to functions (arrow types).
  Commands are analogous to scalars (may be typed $\perp$).
- Extra μ-abstractions and cut account for the fact that we consider commands as functions with scalar values.

On extensionality

- Set theoretic functions are considered extensionally.
- We can use η-expansion to impose extensionality on terms.

Refining η-expansion

- Recall that :

$$s \leftarrow_{\eta} \lambda x\, \mu\alpha \, \langle s \,,\, x \cdot \alpha \rangle \quad .$$

- In front of a product, one can refine :

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

Refining η-expansion

- Recall that :

$$s \leftarrow_{\eta} \lambda x\, \mu\alpha \, \langle s \,, x \cdot \alpha \rangle \quad .$$

- In front of a product, one can refine :

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

- If $e' = t \cdot f$, then :

$$\langle \lambda x\, \mu\alpha \, \langle s \,, e * (x \cdot \alpha) \rangle \,, t \cdot f \rangle \rightarrow^{*} \langle s \,, e * (t \cdot f) \rangle = \left\langle s \,, e * e' \right\rangle \quad .$$

- η-expansion w.r.t. only one component.

Convolution product on contexts (2)

$$\begin{aligned} \langle \lambda z\, s \,,\, e * f \rangle &\leftarrow_{\eta'} \langle \lambda y\, \mu\beta \,\langle \lambda z\, s \,,\, e * (y \cdot \beta) \rangle \,,\, f \rangle \\ &\to \langle \lambda y\, \mu\beta \,\langle \lambda x\, \mu\alpha \,\langle s\,[z := x + y] \,,\, \alpha * \beta \rangle \,,\, e \rangle \,,\, f \rangle \quad . \end{aligned}$$

Convolution product on contexts (2)

$$\begin{aligned}\langle \lambda z\, s\, ,\, e * f \rangle &\leftarrow_{\eta'} \langle \lambda y\, \mu\beta\, \langle \lambda z\, s\, ,\, e * (y \cdot \beta) \rangle\, ,\, f \rangle \\ &\rightarrow \langle \lambda y\, \mu\beta\, \langle \lambda x\, \mu\alpha\, \langle s\, [z := x + y]\, ,\, \alpha * \beta \rangle\, ,\, e \rangle\, ,\, f \rangle \quad .\end{aligned}$$

Compare with :

$$\langle \lambda z\, \varphi(z)\, ,\, e * f \rangle = \langle \lambda y\, \langle \lambda x\, \varphi(x + y)\, ,\, e \rangle\, ,\, f \rangle \quad .$$

Outcome

- This work turns a feature of models of differential λ-calculus into an explicit syntactic operation.
- It turns out syntactic convolution product behaves very similarly to convolution product of distributions.
- In the paper : a system of intersection types, based upon the relational model of LL (cf. [dC07]).

Daniel de Carvalho.
Execution time of λ-terms via non uniform semantics and intersection types.
Manuscript, 2007.

Further work

- Convolution $\bar{\lambda}$-calculus.
- Differential $\bar{\lambda}\mu$-calculus.
- Polarized extension of differential interaction nets.
- ...