Continuations & Co-exponentials

$\lambda\tilde{\lambda}$: a duality of abstraction

Sep 4, HOPE @ ICFP'23

Vikraman Choudhury

Research Associate, Uni. of Glasgow

$\rightsquigarrow$ Marie-Skłodowska-Curie Fellow, Uni. of Bologna

Dualities of Computation

(1) values & continuations (Filinski, Parigot)
(2) call-by-value & call-by-name (Selinger, Wadler)
(3) programs & evaluation contexts (Curien & Herbelin)
(4) producers & consumers in (classical) linear logic (Girard)
(5) senders & receivers in session types (Honda)
(6) polarised adjunction models of System L
(Fiore, Curien, Munch-Maccagnoni)

These are "just" categorical dualities!

Why study Dualities?

(*) Syntactician: fundamental to language design

(*) Semanticist: Dualities in categorical models of computation

Idealy:

Syntactic Dualities $\rightsquigarrow$ Semantic Dualities.

Philosophically:

Dual concepts have the same ontological status!

A micrological study of continuations*

Continuations are functions : $A \to R$

fixed response object R

Fixing R,

$$(-) \to R \cong R^{(-)} : \mathcal{C}^{op} \to \mathcal{C}$$

is self-adjoint (on the right)

(Kock '60, Thielecke '97)

$$\begin{tikzcd}
\mathcal{C}^{op} \arrow[rr, bend right, "R^{(-)}"'] \arrow[rr, phantom, "\perp"] & & \mathcal{C} \arrow[ll, bend right, "R^{(-)}"']
\end{tikzcd}$$

$$\begin{aligned}
&\mathcal{C}^{op}(R^X, Y) \\
\cong\ &\mathcal{C}(Y \times X, R) \\
\cong\ &\mathcal{C}(X \times Y, R) \\
\cong\ &\mathcal{C}(X, R^Y)
\end{aligned}$$

This gives the continuation monad on $\mathcal{C}$.

$$T(X) = R^{R^X} : \mathcal{C} \to \mathcal{C}$$

(* tribute to Melliès)

A micrological study

(*) Every functor $F : \mathcal{C} \to \mathcal{D}$ admits a full-image factorisation by the (bo, ff) factorisation system on $\mathcal{C}at$.

$$
\begin{tikzcd}
\mathcal{C} \arrow[rr, "F"] \arrow[dr, "F_{bo}"'] & & \mathcal{D} \\
& \mathcal{C}_F \arrow[ur, "F_{ff}"'] &
\end{tikzcd}
$$

bijective on objects

fully faithful

- $\mathcal{C}_F(X,Y) \equiv \mathcal{D}(FX, FY)$

- $F_{bo} : \mathcal{C} \longrightarrow \mathcal{C}_F$

  $X \longmapsto X$

  $X \xrightarrow{f} Y \longmapsto FX \xrightarrow{Ff} FY$

- $F_{ff} : \mathcal{C}_F \longrightarrow \mathcal{D}$

  $X \longmapsto FX$

  $X \xrightarrow{f} Y \longmapsto FX \xrightarrow{f} FY$

A micrological study –

If you split the negation functor:

$$
\begin{tikzcd}
\mathcal{C}^{op} \arrow[rr, "R^{(-)}"] \arrow[dr] & & \mathcal{C} \\
& \mathcal{C}_R \simeq \mathcal{C}^{op}_{R^{R^{(-)}}} \arrow[ur] &
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\mathcal{C} \arrow[rr, "R^{(-)}"] \arrow[dr] & & \mathcal{C}^{op} \arrow[dl] \\
& \mathcal{C}^{op}_R \simeq \mathcal{C}_{R^{R^{(-)}}} &
\end{tikzcd}
$$

Call-by name (control category)

Call-by value (co-control category)

category of negated domains (Lafont)

category of continuations

(Hofmann, Streicher, Reus, Agapiev, Moggi ...)

Loch Ness Semantics*

$\mathcal{C}$ biccc with a response object $(0, 1, +, \times, \Rightarrow, R)$

$$
\begin{tikzcd}
\mathcal{C}^{op} \arrow[rr, "R^{(-)}"'] \arrow[dr, "R^{(-)}_{bo}"] \arrow[rr, phantom, bend left=25, "\perp"] & & \mathcal{C} \arrow[ll, bend right=40, "R^{(-)}"'] \\
& \mathcal{C}_R \simeq \mathcal{C}^{op}_{R^{R^{(-)}}} \arrow[ur, "R^{(-)}_{ff}"'] \arrow[ul, bend left=50, "R^{R^{(-)}}"] \arrow[ul, phantom, bend left=20, "\dashv"] &
\end{tikzcd}
$$

(*) $R^{R^{(-)}}$ is a strong monad on $\mathcal{C}$.

(*) $\mathcal{C}_R \simeq \mathcal{C}^{op}_{R^{R^{(-)}}}$, dual of the Kleisli category of $R^{R^{(-)}}$.

(*) $\mathcal{C}_R$ inherits structure from $\mathcal{C}$.

(* from The Blind Spot)

Loch Ness Semantics

Concretely,

(*) $\mathcal{C}$ is cartesian closed : $(-) \times A \dashv (-)^A$

$$\mathcal{C}(C \times A, B) \cong \mathcal{C}(C, B^A)$$

(*) $\mathcal{C}_{R^{R^{(-)}}}$ is co-cartesian co-closed : ${}^{A}(-) \dashv A + (-)$

$$\begin{aligned}
\mathcal{C}_{R^{R^{(-)}}}({}^{A}C, B) &= \mathcal{C}\left(C \times R^A, R^{R^B}\right) \\
&\cong \mathcal{C}\left(C \times R^A \times R^B, R\right) \\
&\cong \mathcal{C}\left(C \times R^{A+B}, R\right) \\
&\cong \mathcal{C}\left(C, R^{R^{A+B}}\right)
\end{aligned}$$

(*)

$$\begin{tikzcd}
\mathcal{C} \arrow[r, bend right, "K"'] \arrow[r, phantom, "T"] & \mathcal{C}_{R^{R^{(-)}}} \arrow[l, bend right, "V"']
\end{tikzcd}$$

is the Kleisli resolution of a strong monad $R^{R^{(-)}}$ on $\mathcal{C}$

Extending Moggi's λ_v

(*) Kleisli exponentials give cbv functions:

$$\mathcal{C}_{R^{R^{(-)}}}(\Gamma \times A, B) \cong \mathcal{C}(\Gamma, \underbrace{A \Rightarrow R^{R^B}})$$

computations — values. — cbv function space

or, double-negated values

or, meta-observations (Taylor)

(*) Coexponentials give binders for covalues (continuation):

Use ${}^{A}\Gamma \equiv \Gamma \times R^{A} \equiv \Gamma \times \widetilde{A}$

$$\mathcal{C}_{R^{R^{(-)}}}(\Gamma \times R^{A}, B) \cong \mathcal{C}_{R^{R^{(-)}}}(\Gamma, \underbrace{R^{R^{A+B}}}).$$

computations — computations — cofunctions

$\lambda\tilde{\lambda}$: a duality of abstraction

Functions are rightist*:

$$\frac{\Gamma, x : A \vdash e : B}{\Gamma \vdash \lambda x . e : A \Rightarrow B} \qquad \frac{\Gamma \vdash e : A \Rightarrow B}{\Gamma, x : A \vdash e\, x : B}$$

These are inverse upto $\beta\eta$ laws.

(in cbv, restricted to values)

(* Levy's terminology)

$\lambda\tilde{\lambda}$: a duality of abstraction

Products are rightist (using fst/snd):

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B}{\Gamma \vdash (a,b) : A \times B} \qquad \frac{\Gamma \vdash p : A \times B}{\Gamma \vdash \mathrm{fst}(p) : A} \qquad \frac{\Gamma \vdash p : A \times B}{\Gamma \vdash \mathrm{snd}(p) : B}$$

Also leftist (using let/pattern matching):

$$\frac{\Gamma, a : A, b : B \vdash c : C}{\Gamma, p : A \times B \vdash \mathrm{let}\ (a,b) = p\ \mathrm{in}\ c : C}$$

$$\frac{\Gamma, p : A \times B \vdash c : C}{\Gamma, a : A, b : B \vdash c[{}^{(a,b)}/_{p}] : C}$$

$\lambda\tilde{\lambda}$: a duality of abstraction

Sums (co-products) are leftist:

$$\frac{\Gamma, a:A \vdash c_1 : C \qquad \Gamma, b:B \vdash c_2 : C}{\Gamma, s:A+B \vdash \mathrm{case}\,(s, a\cdot c_1, b\cdot c_2) : C}$$

$$\frac{\Gamma, s:A+B \vdash c:C}{\Gamma, a:A \vdash c\,[\mathrm{inl}(a)/s]:C} \qquad \frac{\Gamma, s:A+B \vdash c:C}{\Gamma, b:B \vdash c\,[\mathrm{inr}(b)/s]:C}$$

$\lambda\tilde{\lambda}$: a duality of abstraction

With covalues $\tilde{A}$, and $\tilde{\lambda}$, sums become rightist:

$$\frac{\Gamma, x:\tilde{A} \vdash e : B}{\Gamma \vdash \tilde{\lambda}x.e : A+B} \qquad \frac{\Gamma \vdash e : A+B}{\Gamma, x:\tilde{A} \vdash e @ x : B}$$

leftist (inl, inr, case) and rightist ($\tilde{\lambda}$, @) features interact, producing

control effects!

$\lambda\tilde{\lambda}$: typing

functions & cofunctions coexist :

$$\frac{\Gamma, x : A \vdash e : B}{\Gamma \vdash \lambda x . e : A \Rightarrow B} \qquad \frac{\Gamma \vdash e_1 : A \Rightarrow B \qquad \Gamma \vdash e_2 : A}{\Gamma \vdash e_1\, e_2 : B}$$

λ-binding values gives you functions.

$$\frac{\Gamma, x : \tilde{A} \vdash e : B}{\Gamma \vdash \tilde{\lambda} x . e : A + B} \qquad \frac{\Gamma \vdash e_1 : A + B \qquad \Gamma \vdash e_2 : \tilde{A}}{\Gamma \vdash e_1 @ e_2 : B}$$

$\tilde{\lambda}$-binding covalues gives you cofunctions (sums?!)

$\lambda\tilde{\lambda}$: currying & co-currying

The Dual to functions/implication is co-implication/subtraction:

With $\tilde{A}$ types, Define ${}^{A}B / A \leftarrow B / B - A \equiv B \times \tilde{A}$.

Currying & Co-currying co-exist:

$$\text{curry}(f : C \times A \to B) : C \to (A \to B) \equiv \lambda c.\, \lambda a.\, f(c, a)$$

$$\text{uncurry}(f : C \to (A \to B)) : C \times A \to B \equiv \lambda p.\, f(\text{fst}(p))\,(\text{snd}(p))$$

$$\text{cocurry}(f : (A \leftarrow C) \to B) : C \to A + B \equiv \lambda c.\, \tilde{\lambda} \bar{a}.\, f(c, \bar{a})$$

$$\text{councurry}(f : C \to A + B) : (A \leftarrow C) \to B \equiv \lambda p.\, f(\text{fst}(p)) @ \text{snd}(p).$$

$\boxed{\text{Folklore belief: can't make a PL with both due to Degeneracy!}}$

$\lambda\tilde{\lambda}$: algebra of subtraction

$$\text{eval} : (A \to B) \times A \to B \equiv \lambda(f, a).\ f(a)$$

$$\text{uneval} : A \to (B \to A \times B) \equiv \lambda a.\ \lambda b.\ (a, b)$$

$$\text{coeval} : A \to B + (A - B) \equiv \lambda a.\ \tilde{\lambda}\bar{b}.\ (a, \bar{b})$$

$$\text{couneval} : ((B + A) - B) \to A \equiv \lambda(s, \bar{b}).\ s @ \bar{b}.$$

← this is a generalised lem / tnd

$$\text{compose} : (B \to C) \times (A \to B) \to (A \to C) \equiv \lambda(g, f).\ \lambda a.\ g(f(a)).$$

$$\text{cocompose} : (A - C) \to (B - C) + (A - B) \equiv$$

$$\lambda(a, \bar{c}).\ \text{case coeval}(a) \text{ of}$$

$$\text{inl}(b) \mapsto \text{inl}(b, \bar{c})$$

$$\text{inr}(a, \bar{b}) \mapsto \text{inr}(a, \bar{b})$$

$\lambda\tilde{\lambda}$: metaphysical interpretation

Leftist Sums :

$$\frac{\Gamma \vdash e : B}{\Gamma \vdash \mathrm{inr}(e) : A+B}$$

bold choice

Rightist Sums :

$$\frac{\Gamma, x : \tilde{A} \vdash e : B}{\Gamma \vdash \tilde{\lambda} x . e : A+B}$$

locus pusillanimis

pusillanimous choice

e can use x to do "something interesting".

$\lambda\tilde{\lambda}$: classical logic & control

$\text{tnd} : A + \tilde{A} \equiv \tilde{\lambda}\bar{a}.\,\bar{a}$

or

$\text{tnd} : A + (\underline{1} - A) \equiv \text{coeval}\ (*)$

$\tilde{A}\ /\ \underline{1} - A$ is (weak) negation

$\tilde{\lambda}$ is a generalised callCC (with better types).

$$\text{callCC}\,(f : \tilde{A} \to A) : A \equiv \text{case}\ \tilde{\lambda}\bar{a}.\,f(\bar{a})\ \text{of}$$
$$\text{inl}(a) \mapsto a$$
$$\text{inr}(a) \mapsto a$$

or $\text{callCC}(f) = \tilde{\lambda}f\,;\nabla_A$

collapsing nature of callCC (Filinski)

Coapplication is a generalised throw / abort (to the left):

Interaction of inl and @ :

$$\text{throw}(a : A,\ \bar{a} : \breve{A}) : B \;\equiv\; \text{inl}_B(a) \mathbin{@} \bar{a}.$$

$\underline{\tilde{1}}$ is like 0 :

$$\text{abort}(\bar{a} : \breve{1}) : B \;\equiv\; \text{inl}_B(*) \mathbin{@} \bar{a}$$

(*) dne / dni, de Morgan's laws, Pierce's law (call/cc)

(*) Thielecke & Führmann's ftoc / ctof

(*) Curry-Howard : Provability in Gentzen's (S) LK

$\leftrightsquigarrow$

Typability in $\lambda\tilde{\lambda}$

$\lambda\tilde{\lambda}$: equational theory

(*) value restriction for cbv

$\lambda x.e$ are values (frozen), $\tilde{\lambda} x.e$ are not!

(*) theory of substitution & binding :

$$\frac{\Gamma \vdash \theta : \Delta \qquad \Delta \vdash v : A}{\Gamma \vdash \langle \theta, v/x \rangle : \Delta, x:A}$$

$$\left.\begin{aligned} (\lambda x.e)\,v &\equiv e[v/x] \\ \lambda x.v\,x &\equiv v \end{aligned}\right\} \quad \begin{aligned} (\tilde{\lambda} x.e)\,v &\equiv e[v/x] \\ \tilde{\lambda} x.e@x &\equiv e \end{aligned}$$

$\lambda\tilde{\lambda}$: control effects

(*) $\tilde{\lambda}$ - CONST :

$$\frac{\Gamma \vdash e : B}{\Gamma \vdash \tilde{\lambda}x.\, e \equiv \mathrm{inr}(e) : A + B}$$

(*) $\tilde{\lambda}$ - INR - PASS :

$$\frac{\Gamma \vdash e : B}{\Gamma \vdash \tilde{\lambda}x.\, \mathcal{E}[\mathrm{inr}(e) @ x] \equiv \mathrm{inr}(\mathcal{E}[e]) : A + C}$$

(*) $\tilde{\lambda}$ - INL - JUMP :

$$\frac{\Gamma \vdash e : A}{\Gamma \vdash \tilde{\lambda}x.\, \mathcal{E}[\mathrm{inl}(e) @ x] \equiv \mathrm{inl}(e) : A + B}$$

$\lambda\tilde{\lambda}$: non-local exceptions

Fast multiplication (from Harper et. al.):

```
fun mult (l : list int) : int =
    let fun loop [] = 1
          | loop (0::_) = 0
          | loop (h::t) = h * loop t
    in loop l
    end
```

short-circuit

But, $mult\ [1, 2, 0, 3, 4] \rightsquigarrow 1 * (2 * 0)$

$\rightsquigarrow 1 * 0$

$\rightsquigarrow 0$

Using sums for exceptions?

```
fun mult (l: list int): int + int =
    let fun loop [] = inr(1)
          | loop (0 :: _) = inl (0)
          | loop (h::t) = print ("at " ^ tos(h)) ;
                          mapRight (h * _) (loop t)
    in loop l
    end
```

```
mult [1, 2, 0, 3, 4]
~> mapRight (1*_) (mapRight (2*_) (loop [0, 3, 4]))
~> mapRight (1*_) (mapRight (2*_) (inl (0)))
~> mapRight (1*_) (inl (0))
~> inl (0)
```

prints
" at 2 "
" at 1 "

Using $\tilde{\lambda}$:

$$
\begin{array}{l}
\texttt{fun mult (l: list int) : int + int =} \\
\quad \tilde{\lambda} x.\ \texttt{let fun loop [] = } \underline{1} \\
\qquad\qquad \texttt{| loop (0 :: \_) = } \mathrm{inl}(0) \ @\ x \\
\qquad\qquad \texttt{| loop (h :: t) = print ("at " \^{} toS(h));} \\
\qquad\qquad\qquad\qquad\qquad\quad \texttt{h * loop t} \\
\qquad \texttt{in loop l} \\
\qquad \texttt{end}
\end{array}
$$

$$
\begin{array}{l}
\texttt{mult [1, 2, 0, 3, 4]} \\
\rightsquigarrow \tilde{\lambda} x.\ \texttt{loop [1,2,0,3,4]} \\
\rightsquigarrow \tilde{\lambda} x.\ \underline{1} * \texttt{loop [2,0,3,4]} \\
\rightsquigarrow \tilde{\lambda} x.\ \underline{1} * 2 * \texttt{loop [0,3,4]} \\
\rightsquigarrow \tilde{\lambda} x.\ \underline{1} * 2 * \mathrm{inl}(0) \ @\ x \\
\rightsquigarrow \mathrm{inl}\langle 0 \rangle
\end{array}
$$

prints notrace!

$\lambda\tilde{\lambda}$: control effects

(*) CASE - $\tilde{\lambda}$ - β

$$\begin{aligned}&\mathrm{case}\ (\tilde{\lambda}x.\, v)\ \mathrm{of}\\ &\quad \mathrm{inl}(y) \mapsto e_1\\ &\quad \mathrm{inr}(z) \mapsto e_2\end{aligned} \quad \equiv \quad \begin{aligned}&\mathrm{case}\ (\tilde{\lambda}x.\, [v/z]\, e_2)\ \mathrm{of}\\ &\quad \mathrm{inl}(y) \mapsto e_1\\ &\quad \mathrm{inr}(z) \mapsto z\end{aligned}$$

(*) CASE - $\tilde{\lambda}$ - ξ

$$\mathcal{E}\left[\begin{aligned}&\mathrm{case}\ e\ \mathrm{of}\\ &\quad \mathrm{inl}(x) \mapsto e_1\\ &\quad \mathrm{inr}(y) \mapsto e_2\end{aligned}\right] \quad \equiv \quad \begin{aligned}&\mathrm{case}\ e\ \mathrm{of}\\ &\quad \mathrm{inl}(x) \mapsto \mathcal{E}[e_1]\\ &\quad \mathrm{inr}(y) \mapsto \mathcal{E}[e_2].\end{aligned}$$

$\lambda\tilde{\lambda}$: evaluation of tnd

$$\begin{aligned}
&\text{case } (\tilde{\lambda} x . x) \text{ of} && && \text{case } (\tilde{\lambda} x . \underline{1}) \text{ of} \\
&\quad \mathring{\text{inl}}(a) \mapsto 0 && \rightsquigarrow && \quad \mathring{\text{inl}}(a) \mapsto 0 \\
&\quad \mathring{\text{inr}}(\bar{a}) \mapsto \underline{1} && && \quad \mathring{\text{inr}}(z) \mapsto z \\
& && \rightsquigarrow && \text{case } (\mathring{\text{inr}}(\underline{1})) \text{ of} \\
& && && \quad \mathring{\text{inl}}(a) \mapsto 0 \\
& && && \quad \mathring{\text{inr}}(z) \mapsto z \\
& && \rightsquigarrow && \underline{1} .
\end{aligned}$$

$\lambda\tilde{\lambda}$: evaluation of tnd

$$\begin{aligned}
&\mathsf{case}\ (\tilde{\lambda} x . x)\ \mathsf{of} \\
&\quad \mathsf{inl}(a) \mapsto 0 \\
&\quad \mathsf{inr}(\bar{a}) \mapsto \mathsf{inl}(\underline{1}) @ \bar{a}
\end{aligned}
\quad \rightsquigarrow \quad
\begin{aligned}
&\mathsf{case}\ (\tilde{\lambda} x .\ \mathsf{inl}(\underline{1}) @ x)\ \mathsf{of} \\
&\quad \mathsf{inl}(a) \mapsto 0 \\
&\quad \mathsf{inr}(z) \mapsto z
\end{aligned}$$

$$\rightsquigarrow \quad
\begin{aligned}
&\mathsf{case}\ (\mathsf{inl}(\underline{1}))\ \mathsf{of} \\
&\quad \mathsf{inl}(a) \mapsto 0 \\
&\quad \mathsf{inr}(z) \mapsto z
\end{aligned}$$

$$\rightsquigarrow \quad 0 .$$

$\lambda\tilde{\lambda}$: control effects

Complete axiomatisation of control effects :

(*) Indiana theory of control :

Felleisen's control operators

+ Sabry/Felleisen's equations (typed).

(*) Hofmann's equations for call/cc.

$\lambda\tilde{\lambda}$ in action

(*) implement in ML ('a cont, callcc, throw) & Haskell (Cont)

(*) generalised control operators : colam / coapp

(*) non-local exceptions

(*) Double-barrelled backtracking DSL

$$\text{swap} : A + B \xrightarrow{\sim} B + A$$

(*) CPS encoding of effect handlers.

$$\forall r. (Fr \to r) \to (A \to r) \to r$$

$$\cong \forall r. ((Fr + A) \to r) \to r$$

github :

vikraman / coexp

vikraman / agda-coexp

Variations on Duality

Functional Completeness (CCC):

$$\mathcal{C}_{A\times(-)} \simeq \mathcal{C}[x:1 \rightsquigarrow A] \simeq \mathcal{C}_{(-)^A}$$

environment comonad — indeterminate arrow — reader monad

Dual of Functional Completeness (coCCC):

$$\mathcal{C}_{A(-)} \simeq \mathcal{C}[x:A \rightsquigarrow 0] \simeq \mathcal{C}_{A+(-)}$$

handler comonad — indeterminate co-arrow — exception monad

Duality of arrows

λ splits into K/G (Hasegawa):

$$x : \underline{1} \rightsquigarrow C \vdash f : A \rightsquigarrow B \quad \begin{cases} Kx^C.f : C \times A \rightsquigarrow B \\ Gx^C.f : A \rightsquigarrow C \Rightarrow B \end{cases}$$

$\tilde{\lambda}$ splits into $\tilde{K}/\tilde{G}$:

$$x : C \rightsquigarrow 0 \vdash f : A \rightsquigarrow B \quad \begin{cases} \tilde{K}x^C.f : C \Leftarrow A \rightsquigarrow B \\ \tilde{G}x^C.f : A \rightsquigarrow C + B \end{cases}$$

Mechanism for code pointers, breakpoints, debugging

Related ideas:

(*) Filinski's SLC & Griffin's work.

(*) Selinger's control & co-control categories, cbv $\lambda\mu$ + $[\alpha, \beta] / \mu(\alpha, \beta)$

(*) Levy's JwA & Zeilberger's polarities.

(*) Hofmann, Streicher & Reus' semantics for $\lambda\mu$.

(*) Mellies' Dialogue categories

$$C \to (A \parr B)^{**} \cong C \to (A^* \otimes B^*)^* \cong C \otimes A^* \to B^{**}$$

(*) Curien, Fiore, Munch-Maccagnoni's polarised adjunction models

(*) Fiore's inception algebras.

More coexponentials

(*) In linear logic: $T(X) = !(X \multimap R) \multimap R$:

$$C \to T(A \oplus B) \cong C \otimes !(A \multimap R) \longrightarrow T(B).$$

(*) Composable continuations: $T(X) = (X \to S) \to R$

$$C \to T(A+B) \cong C \times (A \to S) \longrightarrow T(B)$$

(*) Quantum Interpretation:

- $c \ominus a = b \iff c = a \vee b$
- Continuations + Quantum?

More on my webpage

Epilogue

(*) higher-order functions $\leftrightsquigarrow$ exponentials.
higher-order continuations $\leftrightsquigarrow$ co-exponentials.

(*) $\lambda\tilde{\lambda}$: Duality of abstraction

- operational, categorical, adequate denotational semantics
- complete axiomatisation of control effects
- algebraic structure of CPS
- implement in ML / Haskell.

(*) More on my website!