

Continuations & Co-exponentials

?? : a duality of abstraction

Sep 4, HOPE @ ICFP'23

Vikraman Choudhury

Research Associate, Uni. of Glasgow

~ Marie-Sklodowska-Curie Fellow, Uni. of Bologna

Dualities of Computation

- (1) values & continuations (Filinski, Parigot)
- (2) call-by-value & call-by-name (Seldinger, 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 F
(Fiore, Curien, Munch-Maccagnoni)

These are "just" categorical Dualities!

Why study Dualities?

- (*) Syntactician: fundamental to language design
 - (*) Semantician: Qualities in categorical models of computation

Ideally

Syntactic Qualities ↳ Semantic Qualities

Philosophically:

Dual concepts have the same ontological status!

A monological study of continuations*

Continuations are functions: $A \rightarrow R$

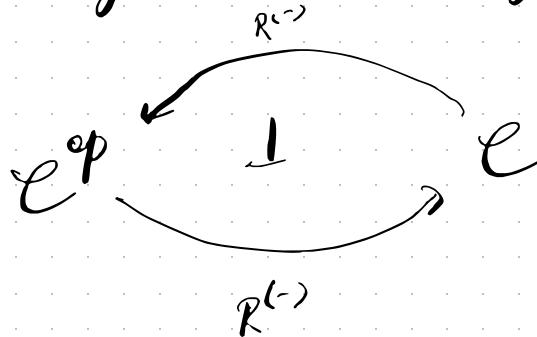
↑

Fixing R ,

fixed response object R

$$(-) \rightarrow R \equiv R^{(-)} : \mathcal{C}^{\text{op}} \rightarrow \mathcal{C} \quad (\text{Kork '60, Thielecke '97})$$

is self-adjoint (on the right)



$$\mathcal{C}^{\text{op}}(R^x, y)$$

$$\cong \mathcal{C}(y \times x, R)$$

$$\cong \mathcal{C}(x \times y, R)$$

$$\cong \mathcal{C}(x, R^y)$$

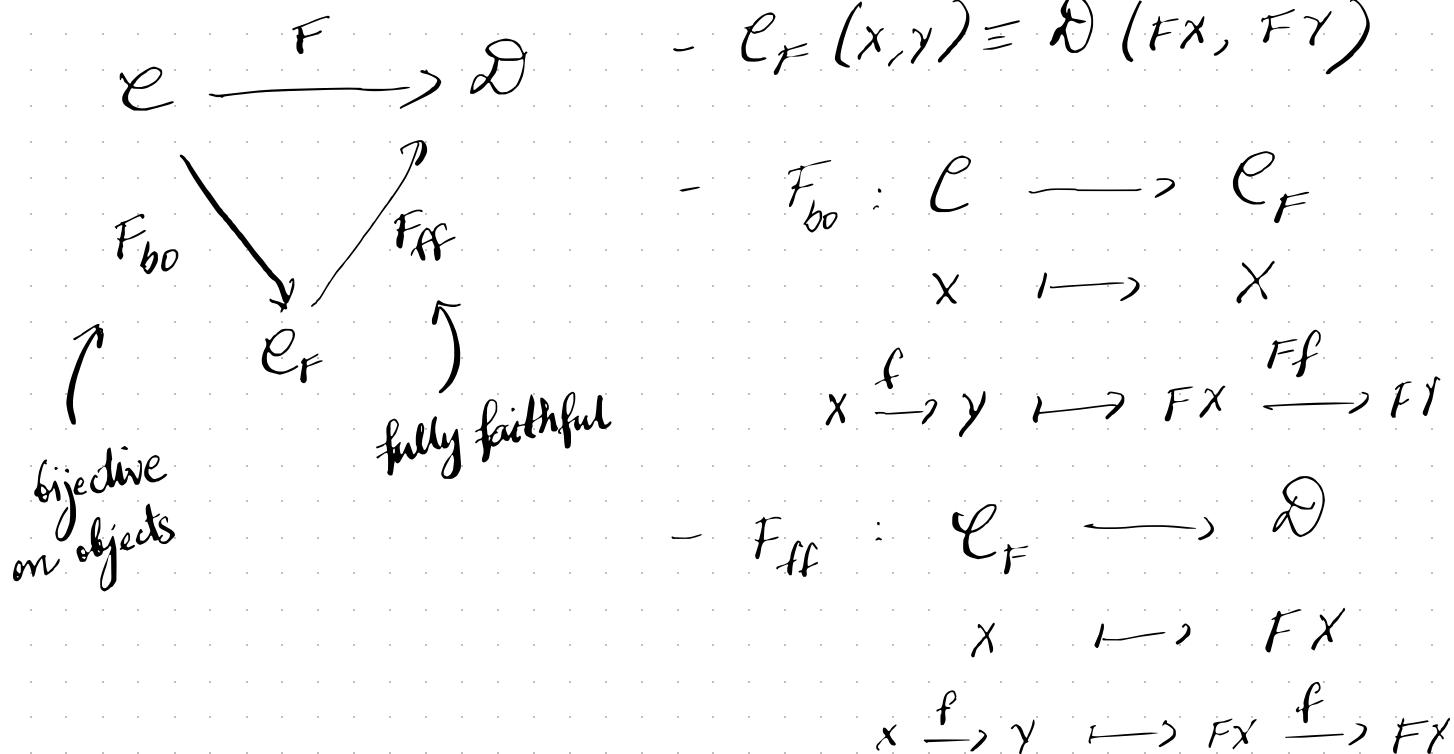
This gives the continuation monad on \mathcal{C} .

$$T(x) = R^{R^x} : \mathcal{C} \rightarrow \mathcal{C}$$

(* tribute to Mellies)

A micrological study

(*) Every functor $F: \mathcal{C} \rightarrow \mathcal{D}$ admits a full image factorisation by the (fo, ff) factorisation system on Cat .



A micrological study

If you split the negation function:

$$\begin{array}{ccc} \mathcal{C}^{\text{op}} & \xrightarrow{R^{(+)}} & \mathcal{C} \\ & \searrow & \swarrow \\ & \mathcal{C}_R \simeq \mathcal{C}_{R^{(+)}}^{\text{op}} & \end{array}$$

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{R^{(+)}} & \mathcal{C}^{\text{op}} \\ & \searrow & \swarrow \\ & \mathcal{C}_R^{\text{op}} \simeq \mathcal{C}_{R^{(+)}} & \end{array}$$

Call-by name

(control
category)

category of negated domains (Lafont)

category of continuations

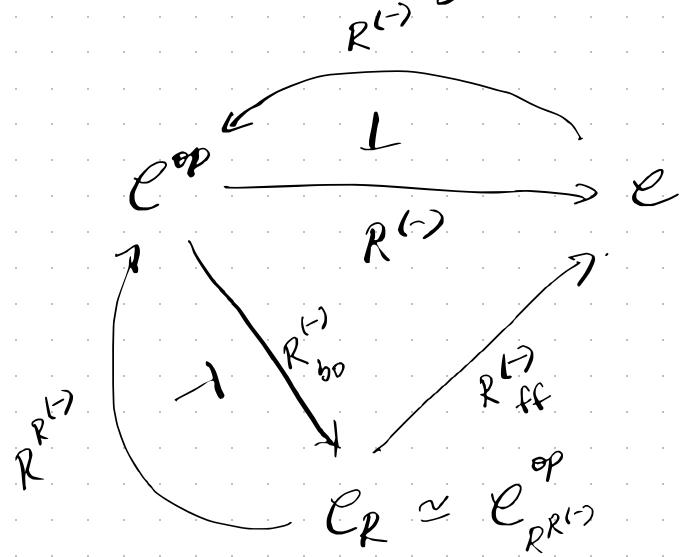
(Hofmann, Streicher, Reus, Agapiev, Moggi)

Call-by value

(co-control
category)

Lockless Semantics*

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



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

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

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

(* from The Blind Spot)

Lack Nexus Semantics

Concretely,

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

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

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

$$\begin{aligned}
 \mathcal{C}_{R^{R(\rightarrow)}}({}^A C, B) &= \mathcal{C}(C \times R^A, R^{RB}) \\
 &\simeq \mathcal{C}(C \times R^A \times R^B, R) \\
 &\simeq \mathcal{C}(C \times R^{A+B}, R) \\
 &\simeq \mathcal{C}(C, R^{R^{A+B}})
 \end{aligned}$$

(*)

$\mathcal{C} \xrightarrow{T} \mathcal{C}_{R^{R(\rightarrow)}}$ is the Kleisli resolution of a strong monad $R^{R(\rightarrow)}$ on \mathcal{C}

Extending Moggi's λ_v

(*) Kleisli exponentials give cbv functions:

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

↑

computations

↑

values

↑

cbv function space

or, double-negated values

or, meta-observations (Taylor)

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

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

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

↑

computations

↑

computations

↑

cofunctions

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

Functions are rightist^{*}:

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

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

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`) :

$$\frac{\Gamma \vdash a:A \quad \Gamma \vdash b:B}{\Gamma \vdash (a,b):A \times B}$$

$$\frac{\Gamma \vdash p:A \times B}{\Gamma \vdash \text{fst}(p):A}$$

$$\frac{\Gamma \vdash p:A \times B}{\Gamma \vdash \text{snd}(p):B}$$

Also leftist (using `let / pattern matching`) :

$$\frac{}{\Gamma, a:A, b:B \vdash c:C}$$

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

$$\frac{}{\Gamma, p:A \times B \vdash c:C}$$

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

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

Sums (co-products) are leftist:

$$\Gamma, a:A \vdash c_1:C$$

$$\Gamma, b:B \vdash c_2:C$$

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

$$\Gamma, s:A+B \vdash c:C$$

$$\Gamma, s:A+B \vdash c:C$$

$$\Gamma, a:A \vdash c[\text{inl}(a)/s] : C$$

$$\Gamma, b:B \vdash c[\text{inr}(b)/s] : C$$

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

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

$$\Gamma, x:\tilde{A} \vdash e:B$$

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

$$\Gamma \vdash e : A+B$$

$$\Gamma, x:\tilde{A} \vdash e@x : B.$$

leftist ($\text{ml}, \text{mr}, \text{case}$) and rightist ($\tilde{\lambda}, @$)

features interact, producing

control effects!

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

functions & cofunctions coexist:

$$\frac{}{\Gamma, x : A \vdash e : B}$$

$$\frac{}{\Gamma \vdash e_1 : A \Rightarrow B}$$

$$\frac{}{\Gamma \vdash e_2 : A}$$

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

$$\frac{}{\Gamma \vdash e_1, e_2 : B}$$

λ -binding values gives you functions.

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

$$\frac{}{\Gamma \vdash e_1 : A + B} \quad \frac{}{\Gamma \vdash e_2 : \tilde{A}}$$

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

$$\frac{}{\Gamma \vdash e_1 @ e_2 : B}$$

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

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

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

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

Currying & Co-currying co-exist:

curry ($f: C \times A \rightarrow B$) : $C \rightarrow (A \rightarrow B) \equiv \lambda c. \lambda a. f(c, a)$

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

co Curry ($f: (A \leftarrow C) \rightarrow B$) : $C \rightarrow A + B \equiv \lambda c. \widetilde{\lambda} \bar{a}. f(c, \bar{a})$

co uncurry ($f: C \rightarrow A + B$) : $(A \leftarrow C) \rightarrow B \equiv \lambda p. f(\text{fst}(p)) @ \text{snd}(p)$.

Folklore belief: can't make a Ph with both due to Degeneracy!

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

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

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

$$\text{coeval} : A \rightarrow B + (A - B) \equiv \lambda a. \tilde{\lambda} \bar{b}. (a, \bar{b}) \leftarrow \text{this is a generalised lem/fnd}$$

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

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

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

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

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

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

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

Leftist Summ:

$$\frac{P \vdash e : B}{P \vdash \text{inv}(e) : A + B}$$

bold choice

Rightist Summ:

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

locus pusillanimis

pusillanimous
choice

e can use x to do "something interesting"

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

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

or

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

$\left\{ \begin{array}{l} \tilde{A} / 1 - A \text{ is} \\ (\text{weak}) \text{ negation} \end{array} \right.$

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

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

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

collapsing nature of callCC
(Finniki)

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

Interaction of inl and $@$:

$$\text{throw}(a : A, \bar{a} : \tilde{A}) : B = \text{inl}_B(a) @ \bar{a}.$$

$\tilde{1}$ is like 0:

$$\text{abort}(\bar{a} : \tilde{1}) : B = \text{inl}_B(\star) @ \bar{a}$$

- (*) De Morgan's laws, Pierce's law (call/cc)
- (*) Thielecke & Führmann's stoc/ctof
- (*) Curry-Howard: Provability in Gentzen's (S) LK
↳
Typeability in $\lambda\tilde{\lambda}$

$\tilde{\lambda}$: equational theory

(*) value restriction for cbv

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

(*) theory of substitution & binding:

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

$$\left. \begin{array}{l} (\lambda x.e)v = e[v/x] \\ \lambda x.vx = v \end{array} \right\} \quad \left. \begin{array}{l} (\tilde{\lambda}x.e)v = e[v/x] \\ \tilde{\lambda}x.e @ x = e \end{array} \right\}$$

λ : control effects

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

$$\Gamma \vdash e : B$$

$$\Gamma \vdash \tilde{\lambda}x. e = \text{inv}(e) : A + B$$

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

$$\Gamma \vdash e : B$$

$$\Gamma \vdash \tilde{\lambda}x. \mathcal{E}[\text{inv}(e) @ x] = \text{inv}(\mathcal{E}[e]) : A + C$$

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

$$\Gamma \vdash e : A$$

$$\Gamma \vdash \tilde{\lambda}x. \mathcal{E}[\text{inl}(e) @ x] = \text{inl}(e) : A + B$$

XX : non-local exceptions

Fast multiplication (from Harpa et al.):

fun mult (l : list int) : int =

let fun loop [] = 1

| loop (0::_) = 0

| loop (n::t) = n * loop t

in loop l

end

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

\rightsquigarrow 1 * 0

\rightsquigarrow 0

short-circuit

Using sums for exceptions?

fun mult (l: list int): int + int =

let fun loop [] = int(1)

| loop (0 :: -) = int(0)

| loop (h :: t) = print ("at " ^ pos(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 *-) (int(0)))

→ mapRight (1 *-) (int(0))

→ int(0)

points
" at 2 "
" at 1 "

Using $\tilde{\lambda}$:

fun mult ($l: \text{list int}$) : $\text{int} + \text{int} =$

$\tilde{\lambda} x.$ let fun loop [$l = 1$

| loop ($0 :: -$) = $\text{int}(0) @ x$

| loop ($n :: t$) = $\text{print}(\text{"at "} \text{at} \text{os}(n));$

$n * \text{loop } t$

in loop l

end

mult [1, 2, 0, 3, 4]

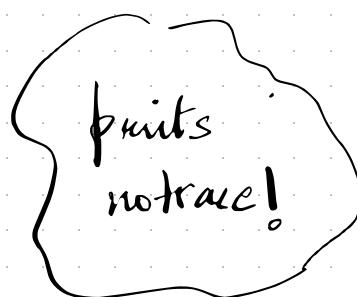
$\rightsquigarrow \tilde{\lambda} x.$ loop [1, 2, 0, 3, 4]

$\rightsquigarrow \tilde{\lambda} x.$ 1 * loop [2, 0, 3, 4]

$\rightsquigarrow \tilde{\lambda} x.$ 1 * 2 * loop [0, 3, 4]

$\rightsquigarrow \tilde{\lambda} x.$ 1 * 2 * $\text{int}(0) @ x$

$\rightsquigarrow \text{int}(0)$



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

(*) CASE - $\tilde{\lambda}\beta$

case $(\tilde{\lambda}x. v)$ of

$\text{inl}(y) \mapsto e_1$

$\text{inr}(z) \mapsto e_2$

\equiv

case $(\tilde{\lambda}x. [v/z]e_2)$ of

$\text{inl}(y) \mapsto e_1$

$\text{inr}(z) \mapsto z$

(*) CASE - $\tilde{\lambda}\xi$

$E \left[\begin{array}{l} \text{case } e \text{ of} \\ \text{inl}(x) \mapsto e_1 \\ \text{inr}(y) \mapsto e_2 \end{array} \right]$

\equiv

case e of

$\text{inl}(x) \mapsto E[e_1]$

$\text{inr}(y) \mapsto E[e_2]$.

λ : evaluation of fnl

case ($\lambda x. x$) of

$\text{int}(a) \mapsto 0$ \rightsquigarrow

$\text{inv}(\bar{a}) \mapsto 1$

case ($\lambda x. 1$) of

$\text{int}(a) \mapsto 0$

$\text{inv}(z) \mapsto z$

\rightsquigarrow case ($\text{inv}(1)$) of

$\text{int}(a) \mapsto 0$

$\text{inv}(z) \mapsto z$

$\rightsquigarrow 1.$

$\lambda\tilde{x}$: evaluation of bind

case ($\lambda x. x$) of

int (a) $\mapsto 0$

inv (a) $\mapsto \text{int}(1) @ \bar{a}$

\rightsquigarrow

case ($\lambda x. \text{int}(1) @ x$) of

int (a) $\mapsto 0$

inv (z) $\mapsto z$

\rightsquigarrow

case (int(1)) of

int (a) $\mapsto 0$

inv (z) $\mapsto z$

$\rightsquigarrow 0.$

27: 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

XX in action

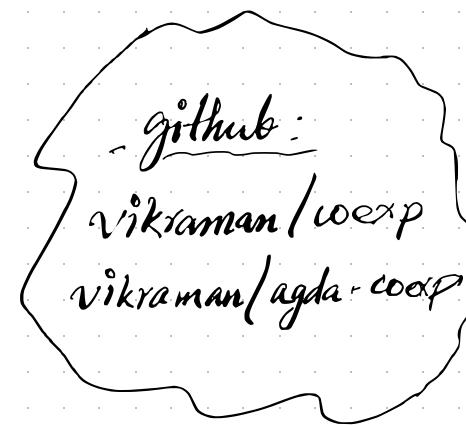
- (*) implement in ML ('a cont, callcc, throw) & Haskell (Cont)
- (*) generalised control operators: comonad/comapp
- (*) non-local exceptions
- (*) Double-barrelled backtracking DSL

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

- (*) CPS encoding of Effect handlers.

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

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



Variations on Duality

Functional Competencies (CCCs)

Dual of Functional Completeness (ccc)

Duality of arrows

λ splits into K/G (Hasegawa):

$$x : 1 \rightsquigarrow C + f : A \rightsquigarrow B$$

$$Kx^c.f : C \times A \rightsquigarrow B$$

$$Gx^c.f : A \rightsquigarrow C \Rightarrow B$$

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

$$x : C \rightsquigarrow 0 + f : A \rightsquigarrow B$$

$$\tilde{K}x^c.f : C \sqsubseteq A \rightsquigarrow B$$

$$\tilde{G}x^c.f : A \rightsquigarrow C + B$$

Mechanism for code pointers,
break points, debugging

Related ideas:

- (*) Tilinški's SLC & Griffán's work.
- (*) Schütter's control & co-control categories, $\text{ctr}[\mu] + [\alpha, \beta] / \mu(\alpha, \beta)$
- (*) Levy's JwA & Zeilberger's polarities.
- (*) Hofmann, Streicher & Reus' semantics for $\Lambda\mu$.
- (*) Mellies' Dialogue categories

$$C \rightarrow (A \otimes B)^{\star\star} \cong C \rightarrow (A^\star \otimes B^\star)^\star \cong C \otimes A^\star \rightarrow B^{\star\star}$$

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

More coexponentials

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

$$C \rightarrow T(A \oplus B) \cong C \otimes ! (A \multimap R) \rightarrow T(B)$$

(*) Composable continuations: $T(x) = (x \rightarrow S) \rightarrow R$

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

(*) Quantum Interpretation:

- $C \otimes a = b \Leftrightarrow C = a \vee b$

- Continuations + Quantum?

More on my webpage

Epilogue

- (*) higher-order functions \rightsquigarrow exponentials.
higher-order continuations \rightsquigarrow ω -exponentials.
- (*) $\lambda\lambda$: 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!