Intersection types for λ-calculi with control

Ugo de'Liguoro

University of Turin, Italy

chocola Lyon - October 1th, 2015

1. The first part of this seminar is an introduction to the λ-calculi with control, and expecially to Streicher and Reus continuation semantics

   (based on Lafont's translation of classical logic into the $\neg\wedge$-fragment of intuitionistic logic).

1. The first part of this seminar is an introduction to the λ-calculi with control, and expecially to Streicher and Reus continuation semantics

   (based on Lafont's translation of classical logic into the $\neg\wedge$-fragment of intuitionistic logic).

2. The second part of the seminar is based on joint work with S. van Bakel and F. Barbanera on intersection types for the $\lambda\mu$-caculus:

   - [BBdL-11] "A Filter Model for $\lambda\mu$", TLCA 2011
   - [BBdL-12] "Characterization of Strongly Normalising $\lambda\mu$-terms", ITRS 2012
   - [dL-??] "The Approximation Theorem for the $\Lambda\mu$-calculus", MSCS to appear
   - [BBdL] "Intersection Types for the $\lambda\mu$-Calculus", submitted

   Draft versions of these papers are available from my page:

   `www.di.unito.it/~deligu`

Concepts

- **control** is the way we decide how to proceed in a program, pointing to the next istruction.
- **continuation** is a point in a program, together with the information on the values of identifiers (the environment) and on what remains to do for completion.
- **control operator** acts on the current continuation, manipulating a representation of the continuation, its *reification*.
- **continuation passing style** is a functional interpretation of control and control operators.

λ-calculi with control

λ-calculi with control

In functional programming languages **control** are explicit constructs dealing with abort or escape, jumps, exceptions, etc.

λ-calculi with control

In functional programming languages **control** are explicit constructs dealing with abort or escape, jumps, exceptions, etc.

But programming and reasoning with continuations and control operators is hard, motivating formalisms based on the λ-calculus:

- $\lambda\mathcal{C}$ and $\lambda\mathcal{F}$ calculi by Felleisen
- $\lambda\mathcal{S}$ calculus for delimited continuations by Danvy and Filinsky ...

λ-calculi with control

In functional programming languages **control** are explicit constructs dealing with abort or escape, jumps, exceptions, etc.

But programming and reasoning with continuations and control operators is hard, motivating formalisms based on the λ-calculus:

- $\lambda\mathcal{C}$ and $\lambda\mathcal{F}$ calculi by Felleisen
- $\lambda\mathcal{S}$ calculus for delimited continuations by Danvy and Filinsky ...

After Griffin's discovery that Felleisen's $\mathcal{C}$-operator can be used to inhabit $\neg\neg$ elimination, also logicians have been working on λ-calculi with control:

- $\lambda\mu$ by Parigot, and its variant $\Lambda\mu$ by de Groote and Saurin
- λcc by Krivine
- $\bar{\lambda}\mu\tilde{\mu}$ by Curien and Herbelin
- $\lambda\mu$tp and $\lambda\mu\hat{\text{tp}}$ by Ariola and Herbelin ...

λ-calculi with control

λ-calculi with control

Definition

A set $\mathcal{E} \subseteq \{C[\,] \mid C[\,] \text{ a context with one hole}\}$ is the set of *evaluation contexts* w.r.t. a notion of reduction $\longrightarrow$ if and only if for all M, if $M \longrightarrow \ldots$ then there are a unique redex R and a unique $E[\,] \in \mathcal{E}$ such that

$$M \equiv E[R]$$

λ-calculi with control

Definition

A set $\mathcal{E} \subseteq \{C[\ ] \mid C[\ ] \text{ a context with one hole}\}$ is the set of *evaluation contexts* w.r.t. a notion of reduction $\longrightarrow$ if and only if for all M, if $M \longrightarrow \ldots$ then there are a unique redex R and a unique $E[\ ] \in \mathcal{E}$ such that

$$M \equiv E[R]$$

The basic idea:

- *continuations* are **evaluation contexts**: $E[\ ]$
- *reification* is some form of **abstraction of an evaluation context**:

$$\lambda x. \ldots E[x] \ldots$$

λ-calculi with control (Felleisen)

Syntax of $\lambda\mathcal{C}$:

$$\begin{array}{llll} \text{Terms:} & M, N & ::= & x \mid \lambda x.M \mid M\,N \mid \mathcal{C}\,M \mid \mathcal{A}\,M \\ \text{Values:} & V & ::= & x \mid \lambda x.M \\ \text{EvConts:} & E & ::= & [\,] \mid E\,M \mid V\,E \end{array}$$

Reduction rules:

$$\begin{array}{llcll} (\beta_v) & E[(\lambda x.M)\,V] & \longrightarrow & E[M[V/x]] & \\ (\mathcal{C}) & E[\mathcal{C}\,M] & \longrightarrow & M(\lambda x.\mathcal{A}E[x]) & x \notin fv(E) \\ (\mathcal{A}) & E[\mathcal{A}\,M] & \longrightarrow & M & \end{array}$$

Note:

$$\mathcal{A}\,M \equiv \mathcal{C}(\lambda\_.M)$$

λ-calculi with control (Felleisen)

$$E[\mathsf{cc}\,(\lambda k.M)] \longrightarrow (\lambda k.k\,M)(\lambda x.\mathcal{A}E[x])$$

cc is definable in terms of $\mathcal{C}$:

$$\mathsf{cc} \equiv \lambda h.\ \mathcal{C}(\lambda f.\, f(h\ f))$$

then

$$\begin{array}{rcl} E[\mathsf{cc}\,(\lambda k.M)] & \longrightarrow & E[\mathcal{C}(\lambda f.\, f((\lambda k.M)\ f))] \\ & \longrightarrow & (\lambda f.\, f((\lambda k.M)\ f))(\lambda x.\mathcal{A}E[x]) \end{array}$$

and

$$\lambda k.k\,M =_{\beta_v} \lambda f.\, f((\lambda k.M)\ f)$$

On the other hand $\mathcal{C}$ is definable in terms of cc and $\mathcal{A}$

$$\mathcal{C}\,M \equiv \mathsf{cc}(\lambda x.\,\mathcal{A}(M\ x))$$

The $\lambda\mu$-calculus: syntax and reduction

$\lambda\mu$ syntax:

$$\begin{array}{rcll} M, N & ::= & x \mid \lambda x.M \mid MN \mid \mu\alpha.Q & \text{(terms)} \\ Q & ::= & [\alpha]M & \text{(commands)} \end{array}$$

Structural substitution:

$$T\{[\alpha]P := [\alpha]PL\} \equiv \text{the replacement of all } [\alpha]N \text{ by } [\alpha]NL \text{ in } T$$

Reduction:

$$\begin{array}{rrcll} (\beta): & (\lambda x.M)N & \longrightarrow & M[N/x] & \\ (\mu): & (\mu\beta.Q)N & \longrightarrow & \mu\beta.Q\{[\beta]P := [\beta]PN\} & \\ (ren): & [\alpha]\mu\beta.Q & \longrightarrow & Q[\alpha/\beta] & \\ (\mu\eta): & \mu\alpha.[\alpha]M & \longrightarrow & M & \text{if } \alpha \notin fn(M) \end{array}$$

A μ-reduction

$$\begin{array}{rl} & \mu\alpha.[\beta](\cdots(\mu\gamma.[\alpha]P)\cdots)N_1\,N_2\,N_3\cdots N_n \\ \longrightarrow_{\lambda\mu} & \mu\alpha.[\beta](\cdots(\mu\gamma.[\alpha]PN_1)\cdots)N_2\,N_3\cdots N_n \\ \longrightarrow_{\lambda\mu} & \mu\alpha.[\beta](\cdots(\mu\gamma.[\alpha]PN_1\,N_2)\cdots)N_3\cdots N_n \\ \overset{*}{\longrightarrow}_{\lambda\mu} & \mu\alpha.[\beta](\cdots(\mu\gamma.[\alpha]PN_1\,N_2\,N_3\cdots N_n)\cdots) \end{array}$$

Taking $E \equiv [\,]N_1\cdots N_n$ this reduction can be written as

$$E[\mu\alpha.[\beta](\cdots(\mu\gamma.[\alpha]P)\cdots)] \overset{*}{\longrightarrow}_{\lambda\mu} \mu\alpha.[\beta](\cdots(\mu\gamma.[\alpha]E[P])\cdots)$$

A μ-reduction

$$\begin{array}{rl} & \mu\alpha.[\beta](\cdots(\mu\gamma.[\alpha]P)\cdots)N_1\,N_2\,N_3\cdots N_n \\ \longrightarrow_{\lambda\mu} & \mu\alpha.[\beta](\cdots(\mu\gamma.[\alpha]PN_1)\cdots)N_2\,N_3\cdots N_n \\ \longrightarrow_{\lambda\mu} & \mu\alpha.[\beta](\cdots(\mu\gamma.[\alpha]PN_1\,N_2)\cdots)N_3\cdots N_n \\ \xrightarrow{*}_{\lambda\mu} & \mu\alpha.[\beta](\cdots(\mu\gamma.[\alpha]PN_1\,N_2\,N_3\cdots N_n)\cdots) \end{array}$$

Taking $E \equiv [\,]N_1\cdots N_n$ this reduction can be written as

$$E[\mu\alpha.[\beta](\cdots(\mu\gamma.[\alpha]P)\cdots)] \xrightarrow{*}_{\lambda\mu} \mu\alpha.[\beta](\cdots(\mu\gamma.[\alpha]E[P])\cdots)$$

- μ-abstraction together with structural substitution are a form of control
- the leading μ cannot be eliminated in general (only in $\mu\alpha.[\alpha]M$ when $\alpha \notin fn(M)$)

The $\Lambda\mu$-calculus (de Groote - Saurin)

Terms:

$$M, N ::= x \mid MN \mid \lambda x.M \mid M\alpha \mid \mu\alpha.M$$

The reduction $M \longrightarrow_{\Lambda\mu} N$ is the compatible closure of the axioms:

$$\begin{array}{lllll}
(\beta_T): & (\lambda x.M)N & \longrightarrow & M[x := N] & \\
(\eta_T): & \lambda x.Mx & \longrightarrow & M & \text{if } x \notin fv(M) \\
(\beta_S): & (\mu\alpha.M)\beta & \longrightarrow & M[\alpha := \beta] & \\
(\eta_S): & \mu\alpha.M\alpha & \longrightarrow & M & \text{if } \alpha \notin fv(M) \\
(fst): & \mu\alpha.M & \longrightarrow & \lambda x.\mu\alpha.M[P\alpha := (Px)\alpha] & \text{if } x \notin fv(M)
\end{array}$$

where

$$M[P\alpha := (PN)\alpha] \equiv \text{replacement of any } P\alpha \text{ by } (PN)\alpha \text{ in } M$$

The $\Lambda\mu$-calculus (de Groote - Saurin)

Parigot's rule

$$(\mu): \ (\mu\alpha.M)N \longrightarrow \mu\alpha.M[P\alpha := (PN)\alpha]$$

is derivable:

$$\begin{array}{lll}
(\mu\alpha.M)N & \longrightarrow_{\Lambda\mu} & (\lambda x.\mu\alpha.M[P\alpha := (Px)\alpha])N \qquad \text{by } (\mathit{fst}) \\
& \longrightarrow_{\Lambda\mu} & (\mu\alpha.M[P\alpha := (Px)\alpha])[x := N] \qquad \text{by } (\beta_T) \\
& \equiv & \mu\alpha.M[P\alpha := (PN)\alpha] \qquad \text{since } x \notin \mathit{fv}(M)
\end{array}$$

Remark: $\mu + \eta_T$ breaks confluence

Recall

$$\begin{array}{lllll} (\eta_T): & \lambda x.Mx & \longrightarrow & M & \text{if } x \notin fv(M) \\ (\mu): & (\mu\alpha.M)N & \longrightarrow & \mu\alpha.M[P\alpha := (PN)\alpha] & \\ (fst): & \mu\alpha.M & \longrightarrow & \lambda x.\mu\alpha.M[P\alpha := (Px)\alpha] & \text{if } x \notin fv(M) \end{array}$$

Without rule (fst), rules (μ) and (η_T) give rise to critical pairs (Py):

$$\mu\alpha.x \;{}_{\eta_T}\!\longleftarrow \lambda y.(\mu\alpha.x)y \longrightarrow_{\mu} \lambda y.\mu\alpha.(x[P\alpha := (Py)\alpha]) \equiv \lambda y.\mu\alpha.x$$

The problem is fixed by rule (fst):

$$\mu\alpha.x \longrightarrow_{fst} \lambda y.\mu\alpha.x$$

Streams (syntax)

A stream is an applicative context of the shape ($k \geqslant 0$):

$$S[\ ] \equiv [\ ]N_1 \dots N_k\beta$$

Putting $\lambda x.M$ in the hole of $S[\ ]$ (for $k > 0$) results into the reduction:

$$(\lambda x.M)N_1N_2 \dots N_k\beta \longrightarrow M[x := N_1]N_2 \dots N_k\beta$$

Instead, putting $\mu\alpha.M$ in $S[\ ]$ results into the reduction:

$$\begin{array}{ll}
 & (\mu\alpha.M)N_1N_2N_3 \dots N_k\beta \\
\xrightarrow{*} & (\mu\alpha.M[P\alpha := (PN_1)\alpha])N_2N_3 \dots N_k\beta \\
\xrightarrow{*} & (\mu\alpha.M[P\alpha := (PN_1)\alpha][P\alpha := (PN_2)\alpha])N_3 \dots N_k\beta \\
 & \dots \\
\xrightarrow{*} & (\mu\alpha.M[P\alpha := (PN_1)\alpha] \dots [P\alpha := (PN_k)\alpha])\beta \\
\longrightarrow & (M[P\alpha := (PN_1)\alpha] \dots [P\alpha := (PN_k)\alpha])[\alpha := \beta] \\
\equiv & (M[\alpha := \beta])[P\beta := (PN_1 \dots N_k)\beta]
\end{array}$$

Continuation Semantics (Streicher-Reus 1998)

Continuation Semantics (Streicher-Reus 1998)

Syntactically continuations are evaluation contexts, that in case of call-by-name have the shape:

$$[\,]M_1 \cdots M_n \qquad \text{with arbitrary } n \in \mathbb{N}$$

Continuation Semantics (Streicher-Reus 1998)

Syntactically continuations are evaluation contexts, that in case of call-by-name have the shape:

$$[\,]M_1 \cdots M_n \qquad \text{with arbitrary } n \in \mathbb{N}$$

If terms denote elements of some domain D, then the space of continuation C should satisfy:

$$C \simeq D \times C \qquad \text{that is} \qquad C \simeq D \times D \times \cdots$$

Continuation Semantics (Streicher-Reus 1998)

Syntactically continuations are evaluation contexts, that in case of call-by-name have the shape:

$$[\,]M_1 \cdots M_n \qquad \text{with arbitrary } n \in \mathbb{N}$$

If terms denote elements of some domain D, then the space of continuation C should satisfy:

$$C \simeq D \times C \qquad \text{that is} \qquad C \simeq D \times D \times \cdots$$

Now D is a space of functions of continuations into some (parametric) domain of results R:

$$D \subseteq C \to R$$

Continuation Semantics (Streicher-Reus 1998)

Continuation domain equations

$$\begin{cases} R & & & \textit{results} \\ D & = & [C \to R] & \textit{denotations} \\ C & = & (D \times C) & \textit{continuations} \end{cases}$$

Continuation Semantics (Streicher-Reus 1998)

Continuation domain equations

$$\begin{cases} R & & & \textit{results} \\ D & = & [C \to R] & \textit{denotations} \\ C & = & (D \times C) & \textit{continuations} \end{cases}$$

If (R, D, C) is a solution then D is an extensional λ-model:

$$D \simeq [C \to R] \simeq [(D \times C) \to R] \simeq [D \to [C \to R]] \simeq [D \to D]$$

The λ-calculus: continuation semantics

Syntax:

$$\text{Trm}: \quad M, N \quad ::= \quad x \mid \lambda x.M \mid MN \qquad \text{(terms)}$$

Semantics: $[\![\cdot]\!]^D : \text{Trm} \to \text{Env} \to D$ where $D = [C \to R]$

$$\begin{aligned}
[\![x]\!]^D\, e\, k &= e\, x\, k \\
[\![\lambda x.M]\!]^D\, e\, \langle d, k\rangle &= [\![M]\!]^D\, e[x := d]\, k \\
[\![MN]\!]^D\, e\, k &= [\![M]\!]^D\, e\, \langle [\![N]\!]^D e, k\rangle
\end{aligned}$$

where $e \in \text{Env} = \mathit{Var} \to D$, $d \in D$ and $k \in C$ so that $[\![M]\!]^D\, e\, k \in R$.

Krivine's machine

Krivine's machine

From

$$
\begin{array}{lcl}
[\![x]\!]^D\, e\, k & = & e\, x\, k \\
[\![\lambda x.M]\!]^D\, e \langle d, k \rangle & = & [\![M]\!]^D\, e[x := d]\, k \\
[\![MN]\!]^D\, e\, k & = & [\![M]\!]^D\, e\, \langle [\![N]\!]^D e, k \rangle
\end{array}
$$

Krivine's machine

From

$$\begin{array}{lcl} [\![x]\!]^D\, e\, k & = & e\, x\, k \\ [\![\lambda x.M]\!]^D\, e\langle d, k\rangle & = & [\![M]\!]^D\, e[x := d]\, k \\ [\![MN]\!]^D\, e\, k & = & [\![M]\!]^D\, e\, \langle [\![N]\!]^D e, k\rangle \end{array}$$

it is possible to reconstruct Krivine's machine:

$$\begin{array}{lcll} (x[e], s) & \to & (e(x), s) & \text{if } x \in dom(e) \\ (\lambda x.M[e], c::s) & \to & (M[e[x := c]], s) & \\ (MN[e], s) & \to & (M[e], N[e]::s) & \end{array}$$

where

$$\begin{array}{llcl} \text{environment :} & e & \in & Var \to_{fin} \text{closure} \\ \text{closure :} & c & ::= & M[e] \\ \text{stack :} & s & ::= & \epsilon \mid c::s \\ \text{state :} & \sigma & ::= & (c, s) \end{array}$$

Interpretation of Parigot's $\lambda\mu$

Semantics: for $e \in \mathsf{Env} = (\mathit{Var} \to D) \times (\mathit{Name} \to C)$ set

$$e\,x = \pi_1(e)(x) \quad x \in \mathit{Var}, \qquad e\,\alpha = \pi_2(e)(\alpha) \quad \alpha \in \mathit{Name}$$

Define: $[\![\cdot]\!]^D : \mathsf{Trm} \to \mathsf{Env} \to D$ and $[\![\cdot]\!]^R : \mathsf{Cmd} \to \mathsf{Env} \to R$

$$\begin{array}{lcl}
[\![x]\!]^D\, e\, k & = & e\, x\, k \\
[\![\lambda x.M]\!]^D\, e\langle d, k\rangle & = & [\![M]\!]^D\, e[x := d]\, k \\
[\![MN]\!]^D\, e\, k & = & [\![M]\!]^D\, e\, \langle [\![N]\!]^D e, k\rangle \\
[\![\mu\alpha.Q]\!]^D\, e\, k & = & [\![Q]\!]^R\, e[\alpha := k] \\
[\![[\alpha]M]\!]^R e & = & [\![M]\!]^D\, e\, (e\,\alpha)
\end{array}$$

Unraveling the definition we get the equation

$$\text{(Swap):} \qquad [\![\mu\alpha.[\beta]M]\!]^D\, e\, k = [\![M]\!]^D\, e[\alpha \mapsto k]\, (e[\alpha \mapsto k]\, \beta)$$

Interpretation of Felleisen's $\lambda\mathcal{C}$ (cbn)

Semantics: $[\![\cdot]\!]^D : \mathsf{Trm} \to \mathsf{Env} \to D$ where $D = [C \to R]$, $e \in \mathsf{Env} = \mathit{Var} \to D$:

$$\begin{array}{lcl} [\![x]\!]^D\, e\, k & = & e\, x\, k \\ [\![\lambda x.M]\!]^D\, e\, \langle d, k\rangle & = & [\![M]\!]^D\, e[x := d]\, k \\ [\![MN]\!]^D\, e\, k & = & [\![M]\!]^D\, e\, \langle [\![N]\!]^D e, k\rangle \\ [\![\mathcal{C}M]\!]^D\, e\, k & = & [\![M]\!]^D\, e\, \langle \mathsf{ret}(k), \mathsf{stop}\rangle \end{array}$$

where

$$\begin{array}{rcl} \mathsf{ret}(k) & = & \lambda\langle d, k'\rangle.\ d\, k \in D \\ \mathsf{stop} & = & \langle \top_D, \mathsf{stop}\rangle = \langle \lambda k.\ \top_R, \mathsf{stop}\rangle = \top_C \in C \end{array}$$

supposing that R has a top element $\top_R$.

Interpreting control operators

$$[\![\mathcal{C}M]\!]^D\, e\, k = [\![M]\!]^D\, e\, \langle \mathsf{ret}(k), \mathsf{stop}\rangle \quad \text{where} \quad \mathsf{ret}(k) = \lambda\langle d, k'\rangle.\ d\ k \in D$$

$$\begin{array}{lcll} [\![\mathcal{A}M]\!]^D\, e\, k & = & [\![\mathcal{C}(\lambda\_.M)]\!]^D\, e\, k & = & [\![M]\!]^D\, e\, \mathsf{stop} \\ [\![\mathsf{cc}\, M]\!]^D\, e\, k & = & [\![\mathcal{C}(\lambda f.f\,(M\,f))]\!]^D\, e\, k & = & [\![M]\!]^D\, e\, \langle \mathsf{ret}(k), k\rangle \end{array}$$

where $f \notin fv(M)$; from which it follows:

$$[\![\mathsf{cc}(\lambda x.\mathcal{A}(M\,x))]\!]^D\, e\, k = [\![M]\!]^D\, e\, \langle \mathsf{ret}(k), \mathsf{stop}\rangle = [\![\mathcal{C}M]\!]^D\, e\, k$$

Defining control operators in $\lambda\mu$

$$\begin{array}{lcl} \mathcal{C} & \equiv & \lambda f.\mu\alpha.[\sigma]f(\lambda x.\mu\beta.[\alpha]x) \\ \mathsf{cc} & \equiv & \lambda f.\mu\alpha.[\alpha]f(\lambda x.\mu\beta.[\alpha]x) \\ \mathcal{A} & \equiv & \lambda f.\mu\alpha.[\sigma]f \end{array}$$

where σ denotes stop. For example:

$$\begin{array}{lcl} [\![(\lambda f.\mu\alpha.[\sigma]f\ M]\!]^D\ e\ k & = & [\![\lambda f.\mu\alpha.[\sigma]f]\!]^D\ e\langle [\![M]\!]^D\ e, k\rangle \\ & = & [\![\mu\alpha.[\sigma]f]\!]^D\ e[f \mapsto [\![M]\!]^D\ e]\ k \\ & = & [\![f]\!]e'[\alpha \mapsto k](e'[\alpha \mapsto k]\ \sigma) \\ & = & [\![M]\!]^D\ e\ \mathsf{stop} \\ & = & [\![\mathcal{A}M]\!]^D\ e\ k \end{array}$$

where $e' = e[f \mapsto [\![M]\!]^D\ e]$ and $e''(\sigma) = \mathsf{stop}$ for all $e'' \in \mathsf{Env}$.

Commands as continuations (BBdL-TLCA'11)

An alternative interpretation of μ-abstraction and commands:

$$[\![\cdot]\!]^D_* : \mathsf{Trm} \to \mathsf{Env} \to D, \qquad [\![\cdot]\!]^C_* : \mathsf{Cmd} \to \mathsf{Env} \to C$$

$$\begin{array}{rcll} [\![\mu\alpha.Q]\!]^D_* \, e \, k & = & d \, k' & \langle d, k' \rangle = [\![Q]\!]^C_* \, e[\alpha := k] \\ [\![[\alpha]M]\!]^C_* e & = & \langle [\![M]\!]^D_* \, e, (e \, \alpha) \rangle & \end{array}$$

Let $M \in \mathsf{Trm}$, $e \in \mathsf{Env}$, $k \in C$ and set $e' = e[\alpha := k]$. Then

$$[\![\mu\alpha.[\beta]M]\!]^D \, e \, k = [\![[\beta]M]\!]^R \, e' = [\![M]\!]^D \, e'(e' \, \beta)$$

on the other hand

$$[\![[\beta]M]\!]^C_* \, e' = \langle [\![M]\!]^D_* \, e', (e' \, \beta) \rangle, \quad \text{so} \quad [\![\mu\alpha.[\beta]M]\!]^D_* \, e \, k = [\![M]\!]^D_* \, e'(e' \, \beta)$$

Therefore $[\![\mu\alpha.[\beta]M]\!]^D = [\![\mu\alpha.[\beta]M]\!]^D_*$ by induction.

Models of $\Lambda\mu$ (Nakazawa, Katsumata)

A stream model is a pair (D, S) that are solutions of the domain equations:

$$D = [S \to D], \qquad S = D \times S.$$

Then for all environment e we define $[\![M]\!]\, e \in D$ by:

$$\begin{array}{lcll}
[\![x]\!]\, e\, s & = & e(x) & \\
[\![\lambda x.M]\!]\, e\langle d, s\rangle & = & [\![M]\!]\, e[x \mapsto d]\, s & \\
[\![MN]\!]\, e\, s & = & [\![M]\!]\, e\langle [\![N]\!]\, e, s\rangle & \\
[\![\mu\alpha.M]\!]\, e\, s & = & [\![M]\!]\, e[\alpha \mapsto s] & \\
[\![M\alpha]\!]\, e\, s & = & ([\![M]\!]\, e\, s)\, e(\alpha) & (d \in D, s \in S)
\end{array}$$

where $e(x) \in D$ and $e(\alpha) \in S$.

Defining $\mathcal{C}$ in $\Lambda\mu$

Consider de Groote's definition:

$$\mathcal{C} \equiv \lambda x.\mu\alpha.x(\lambda f.\ f\ \alpha)$$

and compute

$$\begin{array}{lll} & [\![\mathcal{C}\ M]\!]^D\, e\, s_1\, s_2 & \text{where } D = S \to D \\ = & [\![\mu\alpha.M(\lambda f.\ f\ \alpha)]\!]^D\, e\, s_1\, s_2 & \\ = & [\![M]\!]^D\, e[\alpha \mapsto s_1]\langle [\![\lambda f.\ f\ \alpha]\!]^D\, e[\alpha \mapsto s_1], s_2\rangle & \\ = & [\![M]\!]^D\, e\, \langle [\![\lambda f.\ f\ \alpha]\!]^D\, e[\alpha \mapsto s_1], s_2\rangle & \text{since } \alpha \notin fn(M) \end{array}$$

Observe that

$$[\![\lambda f.\ f\ \alpha]\!]^D\, e[\alpha \mapsto s]\,\langle d, s'\rangle = [\![f]\!]^D e[\alpha \mapsto s, f \mapsto d]\, s' = d\, s$$

hence $[\![\lambda f.\ f\ \alpha]\!]^D\, e[\alpha \mapsto s] = \mathrm{ret}(s)$

Defining $\mathcal{C}$ in $\Lambda\mu$

$$\mathcal{C} \equiv \lambda x.\mu\alpha.x(\lambda f.\ f\ \alpha)$$

$$[\![\mathcal{C}\ M]\!]^D\, e\, s_1\, s_2 = [\![M]\!]^D\, e \langle \mathsf{ret}(s_1), s_2 \rangle$$

Remark: if there exists $\top_D \in D$ then there exists

$$\mathsf{stop} = \top_C = \langle \top_D, \mathsf{stop} \rangle \in C$$

so that

$$[\![\mathcal{C}\ M]\!]^D\, e\ s\ \mathsf{stop} = [\![M]\!]^D\, e \langle \mathsf{ret}(s), \mathsf{stop} \rangle$$

Intersection types

Intersection types

continuation semantics	(Streicher, Reus)	$+$
domain logic	(Abramsky)	$=$
filter model for $\lambda\mu$	(like Barendregt, Coppo, Dezani)	

ω-Algebraic lattices - ω**ALG**

Let $(A, \sqsubseteq)$ be a complete lattice:

- $Z \subseteq A$ is *directed* if $\forall U \subseteq_{fin} Z\, \exists z \in Z.\ \bigsqcup U \sqsubseteq z$
- $x \in A$ is *compact* if $\forall$ directed $Z.\ x \sqsubseteq \bigsqcup Z \Rightarrow \exists z \in Z.\ x \subseteq z$
- A is *algebraic* if $a = \bigsqcup \mathcal{K}(a) = \bigsqcup \{x \sqsubseteq a \mid x \text{ is compact}\}$ for all $a \in A$
- A is *ω-algebraic* if it is algebraic with countable

$$\mathcal{K}(A) = \{x \in A \mid x \text{ is compact}\}$$

ω**ALG** is a category with ω-algebraic lattices as objects and Scott-continuous functions as morphisms. Moreover:

- $A \simeq \mathcal{I}(\mathcal{K}(A))$, the ideal completion of compact points of A
- $f(a) = \bigsqcup \{f(x) \mid x \in \mathcal{K}(a)\}$, continuous functions are determined by their restriction to compact points.

Intersection types: pre-order and filters

Consider an inductively defined language of intersection types:

$$\mathcal{L}: \quad \sigma, \tau ::= \cdots \mid \sigma \wedge \tau \mid \omega$$

Let $(\mathcal{L}, \leqslant)$ be a pre-order s.t.

$$\sigma \wedge \tau = \text{the meet of } \sigma, \tau, \qquad \omega \text{ is the top.}$$

Say that $F \subseteq \mathcal{L}$ is a *filter* if

- $\omega \in F$
- $F \ni \sigma \leqslant \tau \Rightarrow \tau \in F$
- $\sigma, \tau \in F \Rightarrow \sigma \wedge \tau \in F$

Let $\mathcal{F}$ be the set of filters over $(\mathcal{L}, \leqslant)$:

Theorem

*$(\mathcal{F}, \subseteq)$ is in ω**ALG**, with $\uparrow\sigma = \{\tau \in \mathcal{L} \mid \sigma \leqslant \tau\}$ as compact elements.*

Intersection Type Representation of ω-Algebraic Lattices

Theorem (Representation)

Given $A \in \omega\mathbf{ALG}$ *there exists* $(\mathcal{L}_A, \leqslant_A)$ *a pre-ordered intersection type language and a map*

$$\Theta : \mathcal{L}_A \to \mathcal{K}(A) \quad \text{surjective and} \quad \sigma \leqslant_A \tau \Leftrightarrow \Theta(\sigma) \sqsupseteq \Theta(\tau)$$

such that

$$\mathcal{F}^A := \mathit{Filt}(\mathcal{L}_A) \simeq \mathit{Filt}(\mathcal{L}_A/_{\leqslant_A}) \simeq \mathit{Filt}(\mathcal{K}^{op}(A)) \simeq A$$

via the continuous extension of $\uparrow\sigma \mapsto \uparrow\Theta(\sigma)$.

Whenever $\mathcal{F}^A \simeq A$ we say that $(\mathcal{L}_A, \leqslant_A)$ presents A (is a Lindenbaum algebra for A in Abramsky's words).

The inverse limit construction

Consider the (object part of) the co-cone:

$$\begin{array}{lcl} C_0 & = & \{\bot\} \\ D_n & = & [C_n \to R] \\ C_{n+1} & = & D_n \times C_n \end{array}$$

whose limits are:

$$\begin{array}{lcl} D_\infty & = & \lim_n D_n \\ C_\infty & = & \lim_n C_n \end{array}$$

then

$$D_\infty \simeq [C_\infty \to R], \qquad C_\infty \simeq D_\infty \times C_\infty$$

Compact points in D_∞

$\mathcal{K}(D_\infty)$ consists of finite sups of step functions:

$$\bigsqcup_{i \in I} (c_i \Rightarrow r_i), \qquad I \text{ is finite and } c_i \in \mathcal{K}(C_\infty), r_i \in \mathcal{K}(R)$$

where

$$(c \Rightarrow r)(x) = \begin{cases} r & \text{if } c \sqsubseteq x \\ \bot & \text{else} \end{cases}$$

Compact points in C_∞

In general $\mathcal{K}(A \times B) = \mathcal{K}(A) \times \mathcal{K}(B)$, but this holds for finite products!

$$\mathcal{K}(C_\infty) \neq \Pi_n \mathcal{K}(D_\infty) = \text{the infinite product of copies of } \mathcal{K}(D_\infty)$$

let

$$k = \langle d_1, d_2, \ldots \rangle \in \Pi_n \mathcal{K}(D_\infty) \qquad \text{with} \qquad \forall i \in \mathbb{N}.\ d_i \neq \bot$$

then

$$k = \bigsqcup_n k_n \qquad \text{where} \qquad k_n = \langle d_1, \ldots, d_n, \bot, \ldots \rangle$$

but

$$k \not\sqsubseteq k_n \qquad \text{for any } n$$

Theorem

$\mathcal{K}(C_\infty)$ is the subset of all $\langle d_1, d_2, \ldots \rangle \in \Pi_n \mathcal{K}(D_\infty)$ with finitely many $d_i \neq \bot$.

Compact points in C_∞

Otherwise stated

$$C_n = [C_{n-1} \to R] \times [C_{n-2} \to R] \times \cdots \times [C_0 \to R] \times C_0$$

where $C_0 = \{\bot\}$ and $\mathcal{K}(C_n)$ is

$$\mathcal{K}[C_{n-1} \to R] \times \mathcal{K}[C_{n-2} \to R] \times \cdots \times \mathcal{K}[C_0 \to R] \times \mathcal{K}(C_0)$$

so that elements of $\mathcal{K}(C_n)$ have the shape

$$\langle d_1, \ldots, d_{n-1}, \bot \rangle$$

Now

$$\bigcup_n \mathcal{K}(C_n) \simeq \mathcal{K}(C_\infty) \qquad \text{via} \qquad \langle d_1, \ldots, d_{n-1}, \bot \rangle \mapsto \langle d_1, \ldots, d_{n-1}, \bot, \bot, \cdots \rangle$$

Presenting D_∞ and C_∞

Assume that $(\mathcal{L}_R, \leqslant_R)$ is a presentation of R:

$$\begin{array}{lll} \mathcal{L}_D: & \delta ::= & \kappa \to \rho \mid \omega_D \mid \delta \wedge \delta \quad (\rho \in \mathcal{L}_R) \\ \mathcal{L}_C: & \kappa ::= & \delta \times \kappa \mid \omega_C \mid \kappa \wedge \kappa \end{array}$$

Following the characterization of $\mathcal{K}(D_\infty)$ and $\mathcal{K}(C_\infty)$ we expect that (writing $=_A$ for $\leqslant_A \cap \leqslant_A^{-1}$):

$$\begin{array}{ll} \delta =_D \bigwedge_{i\in I} \kappa_i \to \rho_i & \text{or } \delta =_D \omega_D \\ \kappa =_C \bigwedge_{i\in I} (\delta_{i,1} \times \cdots \times \delta_{i,n_i} \times \omega_C) & \text{or } \kappa =_C \omega_C \end{array}$$

which is achieved by postulating

$$\begin{array}{ll} (\kappa \to \delta_1) \wedge (\kappa \to \delta_2) \leqslant_D \kappa \to (\kappa_1 \wedge \kappa_2), & \omega_D \leqslant_D \omega_C \to \omega_R \\ (\delta_1 \times \kappa_1) \wedge (\delta_2 \times \kappa_2) \leqslant_C (\delta_1 \wedge \kappa_2) \times (\kappa_1 \wedge \kappa_2), & \omega_C \leqslant_C \omega_D \times \omega_C \end{array}$$

plus covariance of $\times$ and contravariance/covariance of $\to$.

The filter model

Let $\mathcal{F}_R$, $\mathcal{F}_D$ and $\mathcal{F}_C$ be the set of filters over $(\mathcal{L}_R, \leqslant_R)$, $(\mathcal{L}_D, \leqslant_D)$ and $(\mathcal{L}_C, \leqslant_C)$ resp. ordered by $\subseteq$:

Theorem

By assumption we have $\mathcal{F}_R \simeq R$, so that

$$\mathcal{F}_D \simeq D_\infty \qquad \textit{and} \qquad \mathcal{F}_C \simeq C_\infty$$

therefore

$$\mathcal{F}_D \simeq [\mathcal{F}_C \to \mathcal{F}_R] \qquad \textit{and} \qquad \mathcal{F}_C \simeq \mathcal{F}_D \times \mathcal{F}_C$$

Type judgments

We look for a system deriving statements of the shape:

$$\Gamma \vdash M : \delta \mid \Delta \qquad \Gamma \vdash Q : \rho \mid \Delta$$

where $\Gamma = \{x_1 : \delta_1, \ldots, x_m : \delta_m\}$, $\Delta = \{\alpha_1 : \kappa_1, \ldots, \alpha_n : \kappa_n\}$.

$$\Gamma \models M : \delta \mid \Delta \quad \Leftrightarrow \quad \forall e \in \mathsf{Env}.\ e \models \Gamma, \Delta \Rightarrow [\![M]\!]^D\, e \in [\![\delta]\!]$$

where $[\![\delta]\!] \subseteq D$ and

$$e \models \Gamma, \Delta \quad \Leftrightarrow \quad \forall x \in \mathit{Var}.\ e(x) \in [\![\Gamma(x)]\!] \ \ \& \ \ \forall \alpha \in \mathit{Name}.\ e(\alpha) \in [\![\Delta(\alpha)]\!]$$

and similarly for $\Gamma \vdash Q : \rho \mid \Delta$.

Type interpretation

$$
\begin{array}{lcll}
[\![\nu]\!]^R & \subseteq & R & \nu \in \mathcal{L}_R \text{ a constant} \\
[\![\delta \times \kappa]\!]^C & = & [\![\delta]\!]^D \times [\![\kappa]\!]^C & \\
[\![\kappa \to \rho]\!]^D & = & \{d \in D \mid \forall k \in [\![\kappa]\!]^C .\ d\ k \in [\![\rho]\!]^R\} & \\
[\![\omega]\!]^A & = & A & \text{for } A = R, D, C \\
[\![\sigma \wedge \tau]\!]^A & = & [\![\sigma]\!]^A \cap [\![\tau]\!]^A &
\end{array}
$$

Type interpretation

Theorem

1. $\sigma \leqslant_A \tau \Rightarrow [\![\sigma]\!]^A \subseteq [\![\tau]\!]^A$,
2. $[\![\sigma]\!]^{\mathcal{F}_A} = \{F \in \mathcal{F}_A \mid \sigma \in F\}$ *if* $\mathcal{F}_A \simeq A$ *for* $A = R, D, C$.

Note:

$$\sigma \leqslant \tau \quad \Leftrightarrow \quad \uparrow\tau \subseteq \uparrow\sigma$$

and

$$\sigma \in F \in \mathcal{F} \quad \Leftrightarrow \quad \uparrow\sigma \subseteq F$$

so that

$$[\![M]\!]^{\mathcal{F}_D}\, e \in [\![\delta]\!]^{\mathcal{F}_D} \;\Leftrightarrow\; \delta \in [\![M]\!]^{\mathcal{F}_D}\, e \;\Leftrightarrow\; \uparrow\delta \subseteq [\![M]\!]^{\mathcal{F}_D}\, e$$

Rules from interpretation clauses in $\mathcal{F}$

Rules from interpretation in $\mathcal{F}$

$$\frac{\textit{definiens}, \text{ right hand side of defining clause} = \text{premises}}{\textit{definiendum}, \text{ left hand side of defining clause} = \text{conclusion}}$$

Rules from interpretation clauses in $\mathcal{F}$

Let $e_{\Gamma,\Delta}(x) = \uparrow\Gamma(x)$ and $e_{\Gamma,\Delta}(\alpha) = \uparrow\Delta(\alpha)$, and consider:

$$[\![M]\!]^{\mathcal{F}_D}\, e_{\Gamma,\Delta} \uparrow \kappa = [\![M']\!]^{\mathcal{F}_D}\, e_{\Gamma',\Delta'} \uparrow \kappa'$$

When $\delta = \kappa \to \rho$, noting that

$$\rho \in [\![M]\!]^{\mathcal{F}_D}\, e_{\Gamma,\Delta} \uparrow \kappa \;\Leftrightarrow\; \kappa \to \rho \in [\![M]\!]^{\mathcal{F}_D}\, e_{\Gamma,\Delta}$$

we get the rule

$$\frac{\Gamma' \vdash M' : \kappa' \to \rho \mid \Delta'}{\Gamma \vdash M : \kappa \to \rho \mid \Delta}$$

Rules for λ-abstraction and application

Rules for λ-abstraction and application

Consider the interpretation of $\lambda x.M$ in $\mathcal{F}_D$:

$$[\![\lambda x.M]\!]^{\mathcal{F}_D}\, e\, \langle \uparrow \delta, \uparrow \kappa \rangle = [\![M]\!]^{\mathcal{F}_D}\, e[x :=\uparrow \delta]\ \uparrow \kappa$$

where $\langle \uparrow \delta, \uparrow \kappa \rangle = \uparrow (\delta \times \kappa)$. This reads as the inference rule:

$$\frac{\Gamma, x : \delta \vdash M : \kappa \to \rho \mid \Delta}{\Gamma \vdash \lambda x.M : \delta \times \kappa \to \rho \mid \Delta}$$

Rules for λ-abstraction and application

Consider the interpretation of $\lambda x.M$ in $\mathcal{F}_D$:

$$[\![\lambda x.M]\!]^{\mathcal{F}_D}\, e\langle \uparrow \delta, \uparrow \kappa\rangle = [\![M]\!]^{\mathcal{F}_D}\, e[x :=\uparrow \delta]\ \uparrow \kappa$$

where $\langle \uparrow \delta, \uparrow \kappa\rangle = \uparrow (\delta \times \kappa)$. This reads as the inference rule:

$$\frac{\Gamma, x : \delta \vdash M : \kappa \to \rho \mid \Delta}{\Gamma \vdash \lambda x.M : \delta \times \kappa \to \rho \mid \Delta}$$

Similarly in case of MN, from

$$[\![MN]\!]^{\mathcal{F}_D}\, e\ \uparrow \kappa = [\![M]\!]^{\mathcal{F}_D}\, e\langle [\![N]\!]^{\mathcal{F}_D}\, e, \uparrow \kappa\rangle$$

and for any $\delta \in [\![N]\!]^{\mathcal{F}_D} e$ we get:

$$\frac{\Gamma \vdash M : \delta \times \kappa \to \rho \mid \Delta \quad \Gamma \vdash N : \delta \mid \Delta}{\Gamma \vdash MN : \kappa \to \rho \mid \Delta}$$

Rule for μ-abstraction

The interpretation of $\mu\alpha.Q$ in $\mathcal{F}_D$ is:

$$[\![\mu\alpha.Q]\!]^{\mathcal{F}_D}\ e\ \uparrow\kappa = [\![Q]\!]^{\mathcal{F}_R}\ e[\alpha :=\uparrow\kappa]$$

hence

$$\kappa \to \rho \in [\![\mu\alpha.Q]\!]^{\mathcal{F}_D}\ e \quad \Leftrightarrow \quad \rho \in [\![Q]\!]^{\mathcal{F}_R}\ e[\alpha :=\uparrow\kappa]$$

leading to the rule

$$\frac{\Gamma \vdash Q : \rho \mid \alpha : \kappa, \Delta}{\Gamma \vdash \mu\alpha.Q : \kappa \to \rho \mid \Delta}$$

Rule for commands

On the other hand

$$[\![[\alpha]M]\!]^{\mathcal{F}_R}\, e = [\![M]\!]^{\mathcal{F}_D}\, e\,(e\,\alpha)$$

Since $e = e_{\Gamma,\Delta}$, we have $e\,\alpha = \uparrow \Delta(\alpha)$, where $\Delta(\alpha) = \omega_C$ if $\alpha \notin \mathit{dom}(\Delta)$

Again, if $\Delta(\alpha) = \kappa$, we have

$$\rho \in [\![M]\!]^{\mathcal{F}_D}\, e \uparrow \kappa \;\Leftrightarrow\; \kappa \to \rho \in [\![M]\!]^{\mathcal{F}_D}\, e$$

which leads to the rule

$$\frac{\Gamma \vdash M : \kappa \to \rho \mid \Delta \qquad \Delta(\alpha) = \kappa}{\Gamma \vdash [\alpha]M : \rho \mid \Delta}$$

Intersection type system for $\lambda\mu$

$$\frac{\Gamma(x) = \delta}{\Gamma \vdash x : \delta \mid \Delta} \qquad \frac{A = D, R}{\Gamma \vdash T : \omega_A \mid \Delta}$$

$$\frac{\Gamma, x : \delta \vdash M : \kappa \to \rho \mid \Delta}{\Gamma \vdash \lambda x.M : \delta \times \kappa \to \rho \mid \Delta} \qquad \frac{\Gamma \vdash M : \delta \times \kappa \to \rho \mid \Delta \quad \Gamma \vdash N : \delta \mid \Delta}{\Gamma \vdash MN : \kappa \to \rho \mid \Delta}$$

$$\frac{\Gamma \vdash Q : \rho \mid \alpha : \kappa, \Delta}{\Gamma \vdash \mu\alpha.Q : \kappa \to \rho \mid \Delta} \qquad \frac{\Gamma \vdash M : \kappa \to \rho \mid \Delta \qquad \Delta(\alpha) = \kappa}{\Gamma \vdash [\alpha]M : \rho \mid \Delta}$$

$$\frac{\Gamma \vdash T : \sigma \mid \Delta \qquad \sigma \leqslant_A \tau}{\Gamma \vdash T : \tau \mid \Delta} \qquad \frac{\Gamma \vdash T : \sigma \mid \Delta \qquad \Gamma \vdash T : \tau \mid \Delta}{\Gamma \vdash T : \sigma \wedge \tau \mid \Delta}$$

Invariance under conversion

Theorem (TLCA'11)

Let $=_{\lambda\mu}$ be the convertibility relation induced by $\longrightarrow_{\lambda\mu}$, then:

$$\Gamma \vdash M : \delta \mid \Delta \quad \& \quad M =_{\lambda\mu} N \quad \Rightarrow \quad \Gamma \vdash N : \delta \mid \Delta$$

Proof. In any model D of $\lambda\mu$ we have

$$M =_{\lambda\mu} N \quad \Rightarrow \quad \forall e \in \mathsf{Env}_D.\ [\![M]\!]^D\, e = [\![N]\!]^D\, e$$

but $\mathcal{F}_D \simeq D$ and by construction

$$[\![M]\!]^{\mathcal{F}_D}\, e_{\Gamma,\Delta} = \{\delta \mid \Gamma \vdash M : \delta \mid \Delta\}$$

hence

$$\{\delta \mid \Gamma \vdash M : \delta \mid \Delta\} = \{\delta \mid \Gamma \vdash N : \delta \mid \Delta\}$$

Strong normalisation

Theorem (Pottinger)

A λ-term M is strongly normalisable if and only if there exist Γ, σ such that $\Gamma \vdash M : \sigma$ is derivable in an intersection type system without the type ω.

Can we get rid of ω in the system for $\lambda\mu$?

Of course we have to abandon rule:

$$\frac{}{\Gamma \vdash M : \omega \mid \Delta}\ (\omega)$$

but what about the types?

The meaning of ω

The type

$$\kappa = \delta_1 \times \cdots \times \delta_k \times \omega \in \mathcal{L}_C$$

is semantically inhabited by any infinite tuple:

$$\langle d_1, \ldots, d_k, d_{k+1}, \ldots \rangle \in C$$

such that $d_i \in [\![\delta_i]\!] \subseteq D$ for $i = 1, \ldots, k$.

We then restrict the occurrences of ω to the end of a product type κ
By eliminating rule (ω) the meaning of ω changes:

"lack of information" $\rightsquigarrow$ "partial information about a total object"

The restricted system $\vdash^{-\omega}$

$$\mathcal{L}_R^{-\omega}: \quad \rho ::= \nu \mid \rho \wedge \rho$$
$$\mathcal{L}_C^{-\omega}: \quad \kappa ::= \omega_C \mid \delta \times \kappa \mid \kappa \wedge \kappa \quad (\delta \in \mathcal{L}_D^{-\omega})$$
$$\mathcal{L}_D^{-\omega}: \quad \delta ::= \kappa \to \rho \mid \delta \wedge \delta$$

Note that $\omega_D \notin \mathcal{L}_D^{-\omega}$, so that $\omega_D \times \omega_C \notin \mathcal{L}_C^{-\omega}$.

Definition

$\Gamma \vdash^{-\omega} T : \sigma \mid \Delta$ if it is derivable in the system with all types in $\mathcal{L}_R^{-\omega} \cup \mathcal{L}_C^{-\omega} \cup \mathcal{L}_D^{-\omega}$.

Theorem (ITRS'12)

A $\lambda\mu$ term M is strongly normalisable *if and only if for some Γ, Δ and $\delta \in \mathcal{L}_D^{-\omega}$ it is $\Gamma \vdash^{-\omega} M : \delta \mid \Delta$.*

Intersection Types for Streams

Let (D, S) be a stream model in the category of ω-algebraic lattices

$$\begin{array}{lrcl} \mathcal{L}_T: & \delta & ::= & \varphi \mid \sigma \to \delta \mid \delta \wedge \delta \mid \omega_T \\ \mathcal{L}_S: & \sigma & ::= & \delta \times \sigma \mid \sigma \wedge \sigma \mid \omega_S \end{array}$$

whose intenterpretations are $[\![\delta]\!]^D \subseteq D$ and $[\![\sigma]\!]^S \subseteq S$:

$$\begin{array}{rcll} [\![\varphi]\!]^D & \subseteq & D & \text{fixed for all } \varphi \\ [\![\sigma \to \delta]\!]^D & = & \{d \in D \mid \forall s \in [\![\sigma]\!]^S.\ d(s) \in [\![\delta]\!]^D\} & \\ [\![\delta \times \sigma]\!]^S & = & \{(d :: s) \in S \mid d \in [\![\delta]\!]^D \ \ \& \ \ s \in [\![\sigma]\!]^S\} & \end{array}$$

plus

$$[\![\omega_T]\!]^D = D, \quad [\![\delta_1 \wedge \delta_2]\!]^D = [\![\delta_1]\!]^D \cap [\![\delta_2]\!]^D$$
$$[\![\omega_S]\!]^S = S, \quad [\![\sigma_1 \wedge \sigma_2]\!]^S = [\![\sigma_1]\!]^S \cap [\![\sigma_2]\!]^S$$

Intersection Type Assignment for $\Lambda\mu$

Bases Γ and contexts Δ are defined:

$$\Gamma ::= \varnothing \mid x : \delta, \Gamma \qquad \Delta ::= \varnothing \mid \alpha : \sigma, \Delta$$

Then define the type assignment rules:

$$\frac{}{\Gamma, x : \delta \vdash x : \delta \mid \Delta}$$

$$\frac{\Gamma, x : \delta_1 \vdash M : \sigma \to \delta_2 \mid \Delta}{\Gamma \vdash \lambda x.M : \delta_1 \times \sigma \to \delta_2 \mid \Delta} \qquad \frac{\Gamma \vdash M : \delta_1 \times \sigma \to \delta_2 \mid \Delta \quad \Gamma \vdash N : \delta_1 \mid \Delta}{\Gamma \vdash MN : \sigma \to \delta_2 \mid \Delta}$$

$$\frac{\Gamma \vdash M : \delta \mid \alpha : \sigma, \Delta}{\Gamma \vdash \mu\alpha.M : \sigma \to \delta \mid \Delta} \qquad \frac{\Gamma \vdash M : \sigma \to \delta \mid \Delta \quad \Delta(\alpha) = \sigma}{\Gamma \vdash M\alpha : \delta \mid \Delta}$$

The Filter-Model Theorem

Proposition

1. $\mathcal{F}_D \simeq [\mathcal{F}_S \to \mathcal{F}_D]$ and $\mathcal{F}_S \simeq \mathcal{F}_D \times \mathcal{F}_S$
2. any stream model (D, S) in the category of ω-algebraic lattices is isomorphic to some $(\mathcal{F}_D, \mathcal{F}_S)$.

Let $e \models \Gamma, \Delta$ iff $x : \delta \in \Gamma$ implies $e(x) \in [\![\delta]\!]^D$ and $\alpha : \sigma \in \Delta$ implies $e(\alpha) \in [\![\sigma]\!]^S$; we define:

$$\Gamma \models M : \delta \mid \Delta \Leftrightarrow \forall e \in \mathrm{Env}.\ e \models \Gamma, \Delta \Rightarrow [\![M]\!]^D e \in [\![\delta]\!]^D$$

Theorem (Filter-Model)

$$[\![M]\!]^{\mathcal{F}_D} e = \{\delta \in \mathcal{L}_T \mid \exists \Gamma, \Delta.\ e \models \Gamma, \Delta \ \&\ \Gamma \vdash M : \delta \mid \Delta\}$$

Approximate Normal Forms for $\Lambda\mu$ (Saurin)

Approximate normal forms, ANF, are defined by the grammar:

$$A ::= \Omega \mid \lambda\vec{x}_0\mu\alpha_1\lambda\vec{x}_1 \dots \mu\alpha_n\lambda\vec{x}_n.y\vec{A}_0\beta_1\vec{A}_1 \dots \beta_m\vec{A}_m$$

Any $\Lambda\mu$-term has the shape

$$M \equiv \lambda\vec{x}_0\mu\alpha_1\lambda\vec{x}_1 \dots \mu\alpha_n\lambda\vec{x}_n.R\vec{M}_0\beta_1\vec{M}_1 \dots \beta_m\vec{M}_m$$

where either R is a variable or it is a pre-redex of one of the shapes:

$$(\lambda x.M')N', \quad (\lambda x.M')\alpha, \quad (\mu\alpha.M')N', \quad (\mu\alpha.M')\beta$$

Then we define a mapping $\varphi : \Lambda\mu\text{-terms} \to ANF$:

$$\phi(M) = \begin{cases} \lambda\vec{x}_0\mu\alpha_1\lambda\vec{x}_1 \dots \mu\alpha_n\lambda\vec{x}_n.y\phi(\vec{M}_0)\beta_1\phi(\vec{M}_1) \dots \beta_m\phi(\vec{M}_m) & \text{if } R \equiv y \\ \Omega & \text{else} \end{cases}$$

Approximate Normal Forms for $\Lambda\mu$ (Saurin)

If we define the preorder $\preceq$ over ANF as the least precongruence s.t.:

1. $\Omega \preceq A$,
2. $\lambda x.A\,x \simeq A$, if $x \notin fv(A)$
3. $\mu\alpha.A\,\alpha \simeq A$, if $\alpha \notin fv(A)$
4. $\mu\alpha.A \preceq \lambda x.\mu\alpha.A[P\alpha := (Px)\alpha]$, if $x \notin fv(A)$.

Define the set of *approximants* of M:

$$\mathcal{A}(M) = \{A \in ANF \mid \exists N.\ M \xrightarrow{*} N \ \&\ A \preceq \phi(N)\}$$

Proposition

1. $M \xrightarrow{*} N \ \Rightarrow\ \phi(M) \preceq \phi(N)$
2. if M is closed then $\mathcal{A}(M)$ is an ideal.

Approximation theorem for $\Lambda\mu$

Theorem (MSCS)

$$\Gamma \vdash M : \delta \mid \Delta \Leftrightarrow \exists A \in \mathcal{A}(M).\ \Gamma \vdash A : \delta \mid \Delta$$

By the Filter-Model Theorem we have:

$$\begin{aligned} \delta \in [\![M]\!]^{\mathcal{F}_D} e &\Leftrightarrow \exists \Gamma, \Delta.\ e \models \Gamma, \Delta \ \&\ \Gamma \vdash M : \delta \mid \Delta \\ &\Leftrightarrow \exists \Gamma, \Delta.\ e \models \Gamma, \Delta \ \&\ \exists A \in \mathcal{A}(M).\ \Gamma \vdash A : \delta \mid \Delta \\ &\Leftrightarrow \exists A \in \mathcal{A}(M).\ \delta \in [\![A]\!]^{\mathcal{F}_D} e \end{aligned}$$

and we conclude that

$$[\![M]\!]^{\mathcal{F}_D} e = \bigcup_{A \in \mathcal{A}(M)} [\![A]\!]^{\mathcal{F}_D} e = \bigsqcup_{A \in \mathcal{A}(M)} [\![A]\!]^{\mathcal{F}_D} e$$

Conclusion

Conclusion

- We have seen how a denotational model in a suitable category of algebraic domains with countable basis induces a type theory out of the Lindenbaum algebra of compact points of the domain, and a type assignment system on the basis of the the definition of term denotation. It is interesting to explore what can be done for effects other than continuations, and primarily for side effects.

Conclusion

- We have seen how a denotational model in a suitable category of algebraic domains with countable basis induces a type theory out of the Lindenbaum algebra of compact points of the domain, and a type assignment system on the basis of the the definition of term denotation. It is interesting to explore what can be done for effects other than continuations, and primarily for side effects.
- The semantics we have considered is for call-by-name λ-calculi, but most of the calculi with control have a call-by-value semantics: can we obtain similar results in the call-by-value case? Do we recover known systems of intersection types for call-by-value λ-calculus?

Conclusion

- We have seen how a denotational model in a suitable category of algebraic domains with countable basis induces a type theory out of the Lindenbaum algebra of compact points of the domain, and a type assignment system on the basis of the the definition of term denotation. It is interesting to explore what can be done for effects other than continuations, and primarily for side effects.
- The semantics we have considered is for call-by-name λ-calculi, but most of the calculi with control have a call-by-value semantics: can we obtain similar results in the call-by-value case? Do we recover known systems of intersection types for call-by-value λ-calculus?
- We have mentioned $\lambda\mathcal{F}$ and $\lambda\mathcal{S}$ calculi which capture the notion of "delimited continuation"; what is the proper denotational semantics of such systems? What their intersection type theory?