The Functional Machine Calculus

Willem Heijltjes
University of Bath

TLLA 2021

Rome, 27 June 2021, 2pm

Curry–Howard and effects

The Curry–Howard correspondence

Intuitionistic logic $\leftrightarrow$ Typed λ-calculus

is perfect for **pure** computation. But what about **effects**?

With effects, strategies matter

Call–by–name (CBN) and call–by–value (CBV) give different results:

$$a := 2\,;\,(\lambda x.\,!a)\,(a := 3\,;\,0) \quad \begin{array}{ll} \twoheadrightarrow_{\mathsf{cbn}} & 2 \\ \twoheadrightarrow_{\mathsf{cbv}} & 3 \end{array}$$

Both are desirable behaviours and must be expressible in syntax, locally.

We know many ways how to do this:

- CBV + Thunks (Landin)
- Continuations (Plotkin; Filinski; Curien & Herbelin)
- CBN + Monads (Moggi)
- Kappa-calculus, premonoidal categories, Haskell Arrows
  (Hasegawa; Power, Robinson, Thielecke; Hughes)
- Call–by–push–value & friends
  (Levy; Ehrhard & Guerrieri; Egger, Møgelberg & Simpson)
- Effect handlers (Plotkin & Pretnar)

But these employ a myriad of constructions at type level:

$$A \to TA \quad TTA \to TA \qquad\qquad (\mathcal{C}, \times, \to) \hookrightarrow (\mathcal{M}, \ltimes, \rtimes, I)$$

$$\text{Computation types} \underset{F}{\overset{U}{\rightleftarrows}} \text{Value types}$$

$$(A \to B) \hookrightarrow (A \rightsquigarrow B) \qquad\qquad !A \otimes \underline{B} \qquad \underline{A} \multimap \underline{B}$$

Is this Logic?

The Functional Machine Calculus (FMC)

Two independent modifications to the λ-calculus:

$$M, N \;::=\; x \;\mid\; M\,N \;\mid\; \lambda x.M$$

$$M, N \;::=\; \star \;\mid\; x.\,M \;\mid\; [N]a.\,M \;\mid\; a\langle x\rangle.\,M$$

Sequencing	**Locations**
Split the variable into a unit $\star$ and a variable–with–continuation $x.\,M$	Parameterize abstraction and application in a set of locations $\mathcal{A}$
Encodes **strategies**	Encode **effects**

At the type level: **intuitionistic logic**, but not as we know it. . .

Locations

A simple stack machine/operational semantics (Landin, Krivine)

Stacks: $S \ ::= \ \varepsilon \ | \ S \cdot M$

$$\frac{}{(\varepsilon\ ,\ M)} \qquad \frac{(S\ ,\ M\,N)}{(S \cdot N\ ,\ M)} \qquad \frac{(S \cdot N\ ,\ \lambda x.\, M)}{(S\ ,\ \{N/x\}M)} \qquad \frac{(S\ ,\ x)}{} \qquad \frac{(\varepsilon\ ,\ \lambda x.\, M)}{}$$

- Application: push
- Abstraction: pop and bind to local variable

Basic effects can be encoded with pop and push:

- Input: reading is a pop from a stream
- Output: writing is a push to a stream
- State: cells are stacks of depth one
  update is pop and discard, then push the new value
  lookup is pop and bind, then push the value back

But each effect uses a different stack/stream.

The poly-λ-calculus

Parameterize application and abstraction in a set of locations $\mathcal{A}$

$$M, N \ ::= \ x \ | \ M\,N \ | \ \lambda x.M$$

$$M, N \ ::= \ x \ | \ [N]a.\,M \ | \ a\langle x\rangle.\,M \qquad (a \in \mathcal{A})$$

Embed λ-calculus by a reserved location $\lambda \in \mathcal{A}$ (may omit)

$$\lambda x.\,M \ = \ \lambda\langle x\rangle.\,M \ = \ \langle x\rangle.\,M \qquad M\,N \ = \ [N]\lambda.\,M \ = \ [N].\,M$$

Poly-stack machine/operational semantics

A memory S is a family of stacks: one for every location $a \in \mathcal{A}$.

$$S \;=\; \{S_a \mid a \in \mathcal{A}\} \;=\; S_{a_1}; S_{a_2}; \ldots; S_{a_n}$$

States are pairs (S, M), and transitions are:

$$\frac{(\,S\,;\;S_a\qquad,\;[N]a.\,M\,)}{(\,S\,;\;S_a \cdot N\,,\qquad M\,)} \qquad\qquad \frac{(\,S\,;\;S_a \cdot N\,,\;\;a\langle x\rangle.\,M\,)}{(\,S\,;\;S_a\qquad,\;\{N/x\}M\,)}$$

$$\frac{}{(\,\varepsilon\,,\;M\,)} \qquad\qquad \frac{(\,S\,,\;x\,)}{} \qquad\qquad \frac{(\,S\,;\;\varepsilon_a\,,\;a\langle x\rangle.\,M\,)}{}$$

Encoding state

A memory cell is modelled by a location $c \in \mathcal{A}$

$$\begin{array}{lrcl} \text{update:} & c := N\,;\ M & = & c\langle\_\rangle.\,[N]c.\,M \\ \text{lookup:} & !c & = & c\langle x\rangle.\,[x]c.\,x \end{array}$$

$$c := N\,; M \ : \quad \frac{\dfrac{(\ S\,;\ \varepsilon_c \cdot P\ ,\ c\langle\_\rangle.\,[N]c.\,M\ )}{(\ S\,;\ \varepsilon_c\ ,\ [N]c.\,M\ )}}{(\ S\,;\ \varepsilon_c \cdot N\ ,\ M\ )}$$

$$!c \ : \quad \frac{\dfrac{(\ S\,;\ \varepsilon_c \cdot N\ ,\ c\langle x\rangle.\,[x]c.\,x\ )}{(\ S\,;\ \varepsilon_c\ ,\ [N]c.\,N\ )}}{(\ S\,;\ \varepsilon_c \cdot N\ ,\ N\ )}$$

(Similar to an encoding in π-calculus by Hirschkoff, Prebet & Sangiorgi)

Example

$$a := 2\,;\,(\lambda x.\,!a)\;(a := 3\,;\,0)$$

$$a\langle\_\rangle.\,[2]a.\,[a\langle\_\rangle.\,[3]a.\,0].\,\langle x\rangle.\,a\langle y\rangle.\,[y]a.\,y$$

$$\frac{\frac{\frac{\frac{\frac{\frac{(\;\varepsilon_a\cdot\star\;;\;\varepsilon_\lambda\;,\;a\langle\_\rangle.\,[2]a.\,[a\langle\_\rangle.\,[3]a.\,0].\,\langle x\rangle.\,a\langle y\rangle.\,[y]a.\,y\;)}{(\;\varepsilon_a\;;\;\varepsilon_\lambda\;,\;[2]a.\,[a\langle\_\rangle.\,[3]a.\,0].\,\langle x\rangle.\,a\langle y\rangle.\,[y]a.\,y\;)}}{(\;\varepsilon_a\cdot 2\;;\;\varepsilon_\lambda\;,\;[a\langle\_\rangle.\,[3]a.\,0].\,\langle x\rangle.\,a\langle y\rangle.\,[y]a.\,y\;)}}{(\;\varepsilon_a\cdot 2\;;\;\varepsilon_\lambda\cdot a\langle\_\rangle.\,[3]a.\,0\;,\;\langle x\rangle.\,a\langle y\rangle.\,[y]a.\,y\;)}}{(\;\varepsilon_a\cdot 2\;;\;\varepsilon_\lambda\;,\;a\langle y\rangle.\,[y]a.\,y\;)}}{(\;\varepsilon_a\;;\;\varepsilon_\lambda\;,\;[2]a.\,2\;)}}{(\;\varepsilon_a\cdot 2\;;\;\varepsilon_\lambda\;,\;2\;)}$$

β-Reduction

"Skips" stack actions on other locations:

$$[M]a.\, A_1 \ldots A_n.\, a\langle x\rangle.\, N \quad \longrightarrow \quad A_1 \ldots A_n.\, \{M/x\}N$$

where each A_i is an abstraction or application not on location a

Equivalently: "normal" β-reduction

$$[M]a.\, a\langle x\rangle.\, N \;\longrightarrow\; \{M/x\}N$$

modulo permutations

$$[M]a.\,[N]b.\,P \;\sim\; [N]b.\,[M]a.\,P$$

$$a\langle x\rangle.\,[N]b.\,P \;\sim\; [N]b.\,a\langle x\rangle.\,P \qquad \text{where } x \notin \mathrm{fv}(N)$$

$$a\langle x\rangle.\,b\langle y\rangle.\,P \;\sim\; b\langle y\rangle.\,a\langle x\rangle.\,P$$

Effects are algebraic

(Plotkin & Power)

β-Reduction evaluates state by its algebra:

$$c := M\,;\ c := N\,;\ P \quad = \quad c := N\,;\ P$$

$$c\langle\_\rangle.\ \underline{[M]c.\ c\langle\_\rangle}.\ [N]c.\ P \quad \longrightarrow \quad c\langle\_\rangle.\ [N]c.\ P$$

$$c := M\,;\ !c \quad = \quad c := M\,;\ M$$

$$c\langle\_\rangle.\ \underline{[M]c.\ c\langle x\rangle}.\ [x]c.\ x \quad \longrightarrow \quad c\langle\_\rangle.\ [M]c.\ M$$

Theorem

β-Reduction in the poly-λ-calculus is confluent.

Example

$$a := 2\,;(\lambda x.\,!a)\,(a := 3\,;0)$$

$$\begin{aligned}
& a\langle_\rangle.\,\underline{[2]a}.\,[a\langle_\rangle.\,[3]a.\,0].\,\langle x\rangle.\,\underline{a\langle y\rangle}.\,[y]a.\,y \\
\longrightarrow\; & a\langle_\rangle.\,\underline{[a\langle_\rangle.\,[3]a.\,0].\,\langle x\rangle}.\,[2]a.\,2 \\
\longrightarrow\; & a\langle_\rangle.\,[2]a.\,2 \\
=\; & a := 2\,;2
\end{aligned}$$

Input/output

Dedicated locations in, out $\in \mathcal{A}$ with operations:

$$\begin{aligned} \texttt{read} &= \mathsf{in}\langle x\rangle.\, x \\ \texttt{write}\, N; M &= [N]\mathsf{out}.\, M \end{aligned}$$

Stack machine: initialize with a stream for in:

$$\cdots N_3 \cdot N_2 \cdot N_1$$

β-Reduction: evaluate a term M in the context of an application stream:

$$\ldots [N_3]\mathsf{in}.\, [N_2]\mathsf{in}.\, [N_1]\mathsf{in}.\, M$$

Probabilistic choice and non-determinism encode as special cases of input with locations rnd and nd

(See also Dal Lago, Guerrieri & H.)

Sequencing

CBN and CBV must be expressible in syntax

Need: distinct CBN and CBV translations into poly-λ-calculus

$$a := 2\,;\,(\lambda x.\,!a)\,(a := 3\,;\,0) \quad \begin{array}{ll} \twoheadrightarrow_{\mathsf{cbn}} & a := 2\,;\,2 \\ \twoheadrightarrow_{\mathsf{cbv}} & a := 3\,;\,3 \end{array}$$

$$CBN: \quad a\langle\_\rangle.\,[2]a.\,[a\langle\_\rangle.\,[3]a.\,0].\,\langle x\rangle.\,a\langle y\rangle.\,[y]a.\,y \quad \twoheadrightarrow \quad a\langle\_\rangle.\,[2]a.\,2$$

$$CBV: \quad a\langle\_\rangle.\,[2]a.\,a\langle\_\rangle.\,[3]a.\,[0].\,\langle x\rangle.\,a\langle y\rangle.\,[y]a.\,y \quad \twoheadrightarrow \quad a\langle\_\rangle.\,[3]a.\,3$$

Need: the following to be valid terms

$$[N]: \quad a\langle\_\rangle.\,[3]a.\,[0]$$

$$x.\,M: \quad a\langle\_\rangle.\,[2]a.\,z.\,\langle x\rangle.\,a\langle y\rangle.\,[y]a.\,y$$

The Functional Machine Calculus

Split the variable into a unit $\star$ and a variable–with–continuation $x.\,M$

$$M, N \;::=\; x \;\mid\; M\,N \;\mid\; \lambda x.M$$

$$M, N \;::=\; \star \;\mid\; x.\,M \;\mid\; [N]a.\,M \;\mid\; a\langle x\rangle.\,M$$

Some example terms (the trailing $.\,\star$ will be omitted):

$\langle x\rangle.\,[x].\,[x]$ $\quad$ $\langle x\rangle$ $\quad$ $\langle x\rangle.\,[a\langle y\rangle.\,x.\,[y]a]$ $\quad$ $[\mathrm{rnd}\langle x\rangle.\,[x]\mathrm{out}].\,\langle f\rangle.\,f.\,f.\,f$

Nonsense as functions or λ-terms; fine as stack machine instructions!

Composition $N\,;M$ (or $N.\,M$) has unit $\star$ and is capture-avoiding.

$$
\begin{array}{rcll}
\star\,;M & = & M & \\
x.\,N\,;M & = & x.\,(N\,;M) & \\
[P]a.\,N\,;M & = & [P]a.\,(N\,;M) & \\
a\langle x\rangle.\,N\,;M & = & a\langle y\rangle.\,(\{y/x\}N\,;M) & (y \text{ fresh})
\end{array}
$$

Substitution uses composition for the variable case.

$$
\begin{array}{rcll}
\{M/x\}\star & = & \star & \\
\{M/x\}x.\,N & = & M\,;\{M/x\}N & \\
\{M/x\}y.\,N & = & y.\,\{M/x\}N & (x \neq y) \\
\{M/x\}[P]a.\,N & = & [\{M/x\}P]a.\,\{M/x\}N & \\
\{M/x\}a\langle x\rangle.\,N & = & a\langle x\rangle.\,N & \\
\{M/x\}a\langle y\rangle.\,N & = & a\langle z\rangle.\,\{M/x\}\{z/y\}N & (x \neq y, z \text{ fresh})
\end{array}
$$

β-Reduction skips abstractions and applications but not variables,

$$[M]a.\,A_1 \ldots A_n.\, a\langle x\rangle.\, N \;\longrightarrow\; A_1 \ldots A_n.\,\{M/x\}N$$

where each A_i is of the form $[P]b$ or $b\langle y\rangle$ with $a \neq b$.

Theorem

β-Reduction in the FMC is confluent.

Stack machine/operational semantics

A memory S is a family of stacks: one for every location $a \in \mathcal{A}$.

$$S \;=\; \{S_a \mid a \in \mathcal{A}\} \;=\; S_{a_1}; S_{a_2}; \ldots ; S_{a_n}$$

States are pairs (S, M), and transitions are:

$$\frac{(\,S\,;\;S_a\;,\;[N]a.\,M\,)}{(\,S\,;\;S_a\cdot N\;,\;M\,)} \qquad \frac{(\,S\,;\;S_a\cdot N\;,\;a\langle x\rangle.\,M\,)}{(\,S\,;\;S_a\;,\;\{N/x\}M\,)}$$

$$\frac{}{(\,\varepsilon\;,\;M\,)} \qquad \frac{(\,S\;,\;\star\,)}{} \qquad \frac{(\,S\;,\;x.\,M\,)}{} \qquad \frac{(\,S\,;\;\varepsilon_a\;,\;a\langle x\rangle.\,M\,)}{}$$

Programming in the FMC

Idea: a term M with input S produces output T by a run of the machine

$$\frac{(\,S\,,\,M\,)}{(\,T\,,\,\star\,)}$$

This agrees with composition and identity:

$$\frac{(\,S\,,\,\star\,)}{(\,S\,,\,\star\,)} \qquad\qquad \frac{(\,R\,,\,N\,)}{(\,S\,,\,\star\,)} \wedge \frac{(\,S\,,\,M\,)}{(\,T\,,\,\star\,)} \Rightarrow \frac{(\,R\,,\,N.M\,)}{(\,T\,,\,\star\,)}$$

Primitives

Primitives for arithmetic and Booleans, e.g.

$$\ldots, -2, -1, 0, 1, 2 \ldots \qquad +, \times \qquad \leq, \geq \qquad \bot, \top \qquad \mathbf{if}$$

are defined by their machine transitions, e.g.

$$\frac{(\,S; S_\lambda \cdot 2 \cdot 3\;,\; +\,)}{(\,S; S_\lambda \cdot 5\;,\; \star\,)} \qquad \frac{(\,S; S_\lambda \cdot N \cdot M \cdot \bot\;,\; \mathbf{if}\,)}{(\,S; S_\lambda \cdot N\;,\; \star\,)}$$

$$\frac{(\,S; S_\lambda \cdot 7 \cdot 9\;,\; \geq\,)}{(\,S; S_\lambda \cdot \top\;,\; \star\,)} \qquad \frac{(\,S; S_\lambda \cdot N \cdot M \cdot \top\;,\; \mathbf{if}\,)}{(\,S; S_\lambda \cdot N\;,\; \star\,)}$$

This gives a standard stack calculus, e.g.

$$[1].[2].\, + \,.[3].\, \times$$

evaluates to 9 (returned on the main stack).

Defined operations

Natural operations for state, output, input, and random generation:

$$
\begin{aligned}
\mathsf{get}\ c &= c\langle x\rangle.\,[x]c.\,[x] \\
\mathsf{set}\ c &= \langle x\rangle.\,c\langle \_ \rangle.\,[x]c \\
\mathsf{print} &= \langle x\rangle.\,[x]\mathsf{out} \\
\mathsf{read} &= \mathsf{in}\langle x\rangle.\,[x] \\
\mathsf{rand} &= \mathsf{rnd}\langle x\rangle.\,[x]
\end{aligned}
$$

Definitions (or let-bindings) are redexes, as usual:

$$x = N\,;M \;=\; [N].\,\langle x\rangle.\,M$$

Example:

$$f = (\mathsf{rand}\ .\ \mathsf{set}\ c\ .\ \mathsf{get}\ c)\ .\ f\ .\ f\ .\ +\ .\ \mathsf{print}$$

$$f = (\mathsf{rand} \;.\; \mathsf{set}\ c \;.\; \mathsf{get}\ c) \;.\; f \;.\; f \;.\; + \;.\; \mathsf{print}$$

$$\begin{aligned}
&[\mathsf{rnd}\langle x\rangle.\,[x].\,\langle y\rangle.\,c\langle_\rangle.\,[y]c.\,c\langle z\rangle.\,[z]c.\,[z]].\,\langle f\rangle.\,f.\,f.\;+\,.\,\langle p\rangle.\,[p]out && \rightarrow\\
&[\mathsf{rnd}\langle x\rangle.\,c\langle_\rangle.\,[x]c.\,c\langle z\rangle.\,[z]c.\,[z]].\,\langle f\rangle.\,f.\,f.\;+\,.\,\langle p\rangle.\,[p]out && \rightarrow\\
&[\mathsf{rnd}\langle x\rangle.\,c\langle_\rangle.\;[x]c.\,[x]].\,\langle f\rangle.\,f.\,f.\;+\,.\,\langle p\rangle.\,[p]out && \rightarrow\\
&\mathsf{rnd}\langle x\rangle.\,c\langle_\rangle.\,[x]c.\,[x]\;\;.\;\;\mathsf{rnd}\langle x\rangle.\,c\langle_\rangle.\,[x]c.\,[x]\;\;.\;+\,.\,\langle p\rangle.\,[p]out && \rightarrow\\
&\mathsf{rnd}\langle x\rangle.\,c\langle_\rangle.\,[x]\;\;.\;\;\mathsf{rnd}\langle x\rangle.\,[x]c.\,[x]\;\;.\;+\,.\,\langle p\rangle.\,[p]out &&
\end{aligned}$$

(ε_{out} ;	$S_{rnd} \cdot 6 \cdot 7$;	$\varepsilon_c \cdot \star$;	ε_λ ,	$\mathsf{rnd}\langle x\rangle.\,c\langle\_\rangle.\,[x].\,\mathsf{rnd}\langle x\rangle.\,[x]c.\,[x].\;+\,.\,\langle p\rangle.\,[p]out$)
(ε_{out} ;	$S_{rnd} \cdot 6$;	$\varepsilon_c \cdot \star$;	ε_λ ,	$c\langle\_\rangle.\,[7].\,\mathsf{rnd}\langle x\rangle.\,[x]c.\,[x].\;+\,.\,\langle p\rangle.\,[p]out$)
(ε_{out} ;	$S_{rnd} \cdot 6$;	ε_c ;	ε_λ ,	$[7].\,\mathsf{rnd}\langle x\rangle.\,[x]c.\,[x].\;+\,.\,\langle p\rangle.\,[p]out$)
(ε_{out} ;	$S_{rnd} \cdot 6$;	ε_c ;	$\varepsilon_\lambda \cdot 7$,	$\mathsf{rnd}\langle x\rangle.\,[x]c.\,[x].\;+\,.\,\langle p\rangle.\,[p]out$)
(ε_{out} ;	S_{rnd} ;	ε_c ;	$\varepsilon_\lambda \cdot 7$,	$[6]c.\,[6].\;+\,.\,\langle p\rangle.\,[p]out$)
(ε_{out} ;	S_{rnd} ;	$\varepsilon_c \cdot 6$;	$\varepsilon_\lambda \cdot 7$,	$[6].\;+\,.\,\langle p\rangle.\,[p]out$)
(ε_{out} ;	S_{rnd} ;	$\varepsilon_c \cdot 6$;	$\varepsilon_\lambda \cdot 7 \cdot 6$,	$+\,.\,\langle p\rangle.\,[p]out$)
(ε_{out} ;	S_{rnd} ;	$\varepsilon_c \cdot 6$;	$\varepsilon_\lambda \cdot 13$,	$\langle p\rangle.\,[p]out$)
(ε_{out} ;	S_{rnd} ;	$\varepsilon_c \cdot 6$;	ε_λ ,	$[13]out$)
($\varepsilon_{out} \cdot 13$;	S_{rnd} ;	$\varepsilon_c \cdot 6$;	ε_λ ,	$\star$)

Types

Sequential λ-calculus: "sequencing" but not "locations" (or $\mathcal{A} = \{\lambda\}$)

$$M, N ::= \star \mid x.\,M \mid [N].\,M \mid \langle x\rangle.\,M$$

Types:

$$\rho, \sigma, \tau, \upsilon ::= \sigma_1 \dots \sigma_n \Rightarrow \tau_m \dots \tau_1$$

Idea: a machine run for M: $\sigma_1 \dots \sigma_n \Rightarrow \tau_m \dots \tau_1$

- consumes n inputs of types $\sigma_1 \dots \sigma_n$ from the stack
- produces m outputs of types $\tau_m \dots \tau_1$ on the stack

Examples:

$$\langle x\rangle.\,[x].\,[x] \colon \tau \Rightarrow \tau\tau \qquad \langle x\rangle \colon \tau \Rightarrow \qquad \star \colon (\Rightarrow) \qquad [\star].\,\langle\_\rangle \colon (\Rightarrow)$$

$$\star \colon \rho\sigma\tau \Rightarrow \tau\sigma\rho \qquad \langle f\rangle.\,[\langle x\rangle.\,f.\,[x]] \colon (\sigma \Rightarrow \tau) \Rightarrow (\upsilon\sigma \Rightarrow \tau\upsilon)$$

The type system

Types with vector notation:

$$\rho,\ \sigma,\ \tau,\ \upsilon\ ::=\ \overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau} \qquad \begin{aligned}\overleftarrow{\sigma} &= \sigma_1 \dots \sigma_n \\ \overrightarrow{\sigma} &= \sigma_n \dots \sigma_1\end{aligned}$$

Contexts:

$$\Gamma = \overleftarrow{x} : \overleftarrow{\sigma} = x_1 : \sigma_1, \dots, x_n : \sigma_n$$

The typing rules:

$$\frac{}{\Gamma \vdash \star : \overleftarrow{\tau} \Rightarrow \overrightarrow{\tau}}\ \star \qquad \frac{\Gamma, x : \overleftarrow{\rho} \Rightarrow \overrightarrow{\sigma} \vdash N : \overleftarrow{\sigma}\,\overleftarrow{\tau} \Rightarrow \overrightarrow{\upsilon}}{\Gamma, x : \overleftarrow{\rho} \Rightarrow \overrightarrow{\sigma} \vdash x.\,N : \overleftarrow{\rho}\,\overleftarrow{\tau} \Rightarrow \overrightarrow{\upsilon}}\ \text{var}$$

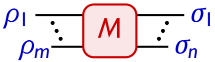
$$\frac{\Gamma, x : \rho \vdash N : \overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}}{\Gamma \vdash \langle x \rangle.\,N : \rho\,\overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}}\ \text{abs} \qquad \frac{\Gamma \vdash M : \rho \qquad \Gamma \vdash N : \rho\,\overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}}{\Gamma \vdash [M].\,N : \overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}}\ \text{app}$$

Example

$$\lambda x.\, xx \;=\; \langle x\rangle.\,[x.\,\star].\,x.\,\star$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x :\, \Rightarrow \vec{\tau} \;\vdash\; \star :\, \overleftarrow{\tau} \Rightarrow \vec{\tau}}\,\star}{x :\, \Rightarrow \vec{\tau} \;\vdash\; x.\,\star :\, \Rightarrow \vec{\tau}}\,\text{var} \qquad \dfrac{\dfrac{}{x :\, \Rightarrow \vec{\tau} \;\vdash\; \star :\, \overleftarrow{\tau}(\Rightarrow \vec{\tau}) \Rightarrow (\Rightarrow \vec{\tau})\vec{\tau}}\,\star}{x :\, \Rightarrow \vec{\tau} \;\vdash\; x.\,\star :\, (\Rightarrow \vec{\tau}) \Rightarrow (\Rightarrow \vec{\tau})\vec{\tau}}\,\text{var}}{x :\, \Rightarrow \vec{\tau} \;\vdash\; [x.\,\star].\,x.\,\star :\, \Rightarrow (\Rightarrow \vec{\tau})\vec{\tau}}\,\text{app}}{\vdash\; \langle x\rangle.\,[x.\,\star].\,x.\,\star :\, (\Rightarrow \vec{\tau}) \Rightarrow (\Rightarrow \vec{\tau})\vec{\tau}}\,\text{abs}$$

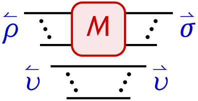
String diagrams



$$M\colon \rho_1 \ldots \rho_m \Rightarrow \sigma_n \ldots \sigma_1$$

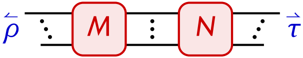
Expansion:

$$\frac{\Gamma \vdash M\colon \overleftarrow{\rho} \Rightarrow \overleftarrow{\sigma}}{\Gamma \vdash M\colon \overleftarrow{\rho}\overleftarrow{\upsilon} \Rightarrow \overrightarrow{\upsilon}\overrightarrow{\sigma}}$$



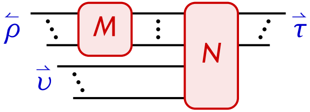
Strict composition:

$$\frac{\Gamma \vdash M\colon \overleftarrow{\rho} \Rightarrow \overrightarrow{\sigma} \qquad \Gamma \vdash N\colon \overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}}{\Gamma \vdash M.N\colon \overleftarrow{\rho} \Rightarrow \overrightarrow{\tau}}$$

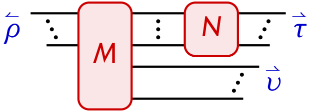


Composition

$$\frac{\Gamma \vdash M : \overleftarrow{\rho} \Rightarrow \overrightarrow{\sigma} \qquad \Gamma \vdash N : \overleftarrow{\sigma}\overleftarrow{\upsilon} \Rightarrow \overrightarrow{\tau}}{\Gamma \vdash M.N : \overleftarrow{\rho}\overleftarrow{\upsilon} \Rightarrow \overrightarrow{\tau}}$$



$$\frac{\Gamma \vdash M : \overleftarrow{\rho} \Rightarrow \overrightarrow{\upsilon}\overrightarrow{\sigma} \qquad \Gamma \vdash N : \overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}}{\Gamma \vdash M.N : \overleftarrow{\rho} \Rightarrow \overrightarrow{\upsilon}\overrightarrow{\tau}}$$



As a partial operation on types:

$$\frac{\Gamma \vdash M : \sigma \qquad \Gamma \vdash N : \tau}{\Gamma \vdash M.N : \sigma \cdot \tau}$$

$$(\overleftarrow{\rho} \Rightarrow \overrightarrow{\sigma}) \cdot (\overleftarrow{\sigma}\,\overleftarrow{\upsilon} \Rightarrow \overrightarrow{\tau}) \quad = \quad (\overleftarrow{\rho}\,\overleftarrow{\upsilon} \Rightarrow \overrightarrow{\tau})$$

$$(\overleftarrow{\rho} \Rightarrow \overrightarrow{\upsilon}\,\overrightarrow{\sigma}) \cdot (\overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}) \quad = \quad (\overleftarrow{\rho} \Rightarrow \overrightarrow{\upsilon}\,\overrightarrow{\tau})$$

Termination

Vectors type stacks:

$$(S : \vec{\sigma}) \cdot (M : \tau) \;=\; (S \cdot M) : \vec{\sigma}\,\tau$$

Idea:

$$M : \overleftarrow{\sigma} \Rightarrow \vec{\tau} \quad\Longrightarrow\quad \forall S : \vec{\sigma}.\ \exists T : \vec{\tau}.\ \ \frac{(\,S\ ,\ M\,)}{(\,T\ ,\ \star\,)}$$

Proved directly by induction on derivations.

Gives a basic, intuitive, reducibility-style proof. (Tait, Girard)

Theorem

For typed FMC-terms the machine terminates.

Poly-types

Parameterize inputs and outputs in locations $\mathcal{A}$:

$$\begin{aligned} \rho,\ \sigma,\ \tau,\ \upsilon &::= \overleftarrow{\sigma}_{\mathcal{A}} \Rightarrow \overrightarrow{\tau}_{\mathcal{A}} \\ \overrightarrow{\tau}_{\mathcal{A}} &::= \{\overrightarrow{\tau}_a \mid a \in \mathcal{A}\} \\ \overrightarrow{\tau} &::= \tau_n \ldots \tau_1 \end{aligned}$$

Notation:

Concatenation $\overrightarrow{\sigma}_{\mathcal{A}}\overrightarrow{\tau}_{\mathcal{A}}$: $\quad \overrightarrow{\sigma}_{\mathcal{A}}\overrightarrow{\tau}_{\mathcal{A}} = \{\overrightarrow{\sigma}_a\overrightarrow{\tau}_a \mid a \in \mathcal{A}\}$

Injection $a(\overrightarrow{\tau})$: $\quad a(\overrightarrow{\tau})_a = \overrightarrow{\tau}$ and $a(\overrightarrow{\tau})_b = \varepsilon$ where $a \neq b$

Slicing τ_a: $\quad (\overleftarrow{\sigma}_{\mathcal{A}} \Rightarrow \overrightarrow{\tau}_{\mathcal{A}})_a = \overleftarrow{\sigma}_a \Rightarrow \overrightarrow{\tau}_a$

Composition $\sigma \cdot \tau$: $\quad \sigma \cdot \tau = \{\sigma_a \cdot \tau_a \mid a \in \mathcal{A}\}$

$$\frac{}{\Gamma \vdash \star : \overleftarrow{\tau}_{\mathcal{A}} \Rightarrow \overrightarrow{\tau}_{\mathcal{A}}}\ \star$$

$$\frac{\Gamma, x : \overleftarrow{\rho}_{\mathcal{A}} \Rightarrow \overrightarrow{\sigma}_{\mathcal{A}} \vdash N : \overleftarrow{\sigma}_{\mathcal{A}}\ \overleftarrow{\tau}_{\mathcal{A}} \Rightarrow \overrightarrow{\upsilon}_{\mathcal{A}}}{\Gamma, x : \overleftarrow{\rho}_{\mathcal{A}} \Rightarrow \overrightarrow{\sigma}_{\mathcal{A}} \vdash x.\,N : \overleftarrow{\rho}_{\mathcal{A}}\ \overleftarrow{\tau}_{\mathcal{A}} \Rightarrow \overrightarrow{\upsilon}_{\mathcal{A}}}\ \mathsf{var}$$

$$\frac{\Gamma \vdash M : \rho \qquad \Gamma \vdash N : a(\rho)\ \overleftarrow{\sigma}_{\mathcal{A}} \Rightarrow \overrightarrow{\tau}_{\mathcal{A}}}{\Gamma \vdash [M]a.\,N : \overleftarrow{\sigma}_{\mathcal{A}} \Rightarrow \overrightarrow{\tau}_{\mathcal{A}}}\ \mathsf{app}$$

$$\frac{\Gamma, x : \rho \vdash N : \overleftarrow{\sigma}_{\mathcal{A}} \Rightarrow \overrightarrow{\tau}_{\mathcal{A}}}{\Gamma \vdash a\langle x\rangle.\,N : a(\rho)\ \overleftarrow{\sigma}_{\mathcal{A}} \Rightarrow \overrightarrow{\tau}_{\mathcal{A}}}\ \mathsf{abs}$$

Example

$$f = (\text{rand}.\ \text{set}\ c.\ \text{get}\ c).\ f.\ f.\ +\ .\ \text{print}$$

$$\begin{array}{rcrcl} & & + & : & \mathbb{Z}\ \mathbb{Z} \Rightarrow \mathbb{Z} \\ \text{rand} & = & \text{rnd}\langle x\rangle.\ [x] & : & \text{rnd}(\mathbb{Z}) \Rightarrow \mathbb{Z} \\ \text{print} & = & \langle x\rangle.\ [x]\text{out} & : & \mathbb{Z} \Rightarrow \text{out}(\mathbb{Z}) \\ \text{set}\ c & = & \langle x\rangle.\ c\langle \_\rangle.\ [x]c & : & \mathbb{Z}\ c(\mathbb{Z}) \Rightarrow c(\mathbb{Z}) \\ \text{get}\ c & = & c\langle x\rangle.\ [x]c.\ [x] & : & c(\mathbb{Z}) \Rightarrow c(\mathbb{Z})\ \mathbb{Z} \end{array}$$

$$\begin{array}{ll} \text{rand}.\ \text{set}\ c & :\ \text{rnd}(\mathbb{Z})\ c(\mathbb{Z}) \ \Rightarrow\ c(\mathbb{Z}) \\ \text{rand}.\ \text{set}\ c.\ \text{get}\ c & :\ \text{rnd}(\mathbb{Z})\ c(\mathbb{Z}) \ \Rightarrow\ c(\mathbb{Z})\ \mathbb{Z} \\ f = (\text{rand}.\ \text{set}\ c.\ \text{get}\ c) & :\ (\Rightarrow) \\ f = (\text{rand}.\ \text{set}\ c.\ \text{get}\ c).\ f.\ f & :\ \text{rnd}(\mathbb{Z}\ \mathbb{Z})\ c(\mathbb{Z}) \ \Rightarrow\ c(\mathbb{Z})\ \mathbb{Z}\ \mathbb{Z} \\ f = (\text{rand}.\ \text{set}\ c.\ \text{get}\ c).\ f.\ f.\ +\ .\ \text{print} & :\ \text{rnd}(\mathbb{Z}\ \mathbb{Z})\ c(\mathbb{Z}) \ \Rightarrow\ c(\mathbb{Z})\ \text{out}(\mathbb{Z}) \end{array}$$

Semantics

(with Chris Barrett)

As a category

Objects are type vectors $\overleftarrow{\tau}$

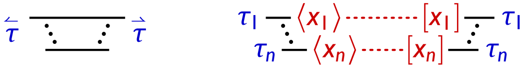
Morphisms in $\overleftarrow{\sigma} \longrightarrow \overleftarrow{\tau}$ are terms $M\colon \overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}$ modulo:

α-equivalence	$\Longrightarrow$	Category
$\alpha\beta\eta$-equivalence	$\Longrightarrow$	Premonoidal Category
"machine equivalence"	$\Longrightarrow$	Cartesian Closed Category

η-Equivalence

$$M =_\eta \langle x\rangle.\,[x].\,M \qquad \text{if } x \notin \mathrm{fv}(M)$$

For $\overleftarrow{\tau} = \tau_1 \dots \tau_n$ let $\langle\overleftarrow{x}\rangle = \langle x_1\rangle \dots \langle x_n\rangle$ and $[\overrightarrow{x}] = [x_n] \dots [x_1]$

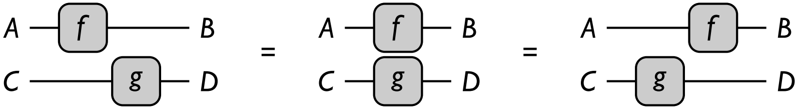


$$\star\colon \overleftarrow{\tau} \Rightarrow \overrightarrow{\tau} \qquad =_\eta \qquad \langle\overleftarrow{x}\rangle.\,[\overrightarrow{x}]\colon \overleftarrow{\tau} \Rightarrow \overrightarrow{\tau}$$

Premonoidal categories

(Power & Robinson, Power & Thielecke)

Monoidal: a bifunctor $- \otimes -$



Premonoidal: functors $- \otimes A$ and $A \otimes -$

Theorem

$\alpha\beta\eta$-Equivalent typed FMC-terms form a strict premonoidal category.

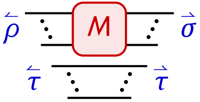
Premonoidal product:

$$\overleftarrow{\sigma} \otimes \overleftarrow{\tau} \;=\; \overleftarrow{\sigma}\overleftarrow{\tau}$$

(associator and unitor are identities)

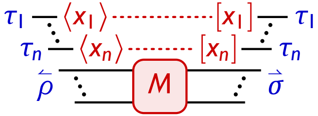
Actions on morphisms: for $M\colon \overleftarrow{\rho} \Rightarrow \overrightarrow{\sigma}$

$M \otimes \overleftarrow{\tau}$



$M\colon \overleftarrow{\rho}\overleftarrow{\tau} \Rightarrow \overrightarrow{\tau}\overrightarrow{\sigma}$

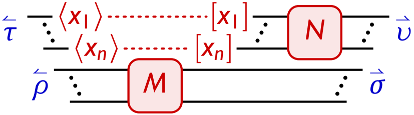
$\overleftarrow{\tau} \otimes M$



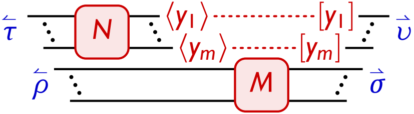
$\langle \overleftarrow{x} \rangle . M . [\overrightarrow{x}]\colon \overleftarrow{\tau}\overleftarrow{\rho} \Rightarrow \overrightarrow{\sigma}\overrightarrow{\tau}$

$\alpha\beta\eta$-Equivalent terms are not monoidal:

$\langle\overleftarrow{x}\rangle.\,M.\,[\overrightarrow{x}].\,N:\ \overleftarrow{\tau}\overleftarrow{\rho}\Rightarrow\overrightarrow{\sigma}\overrightarrow{\upsilon}$



$N.\,\langle\overleftarrow{y}\rangle.\,M.\,[\overrightarrow{y}]:\ \overleftarrow{\tau}\overleftarrow{\rho}\Rightarrow\overrightarrow{\sigma}\overrightarrow{\upsilon}$



In particular, *M* and *N* can be variables (or primitives)

Machine equivalence

(cf. Pitts & Stark 1998)

Typed terms are machine equivalent

$$M \sim M' : \overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}$$

if equivalent inputs give equivalent outputs

$$\forall\, S \sim S' : \overrightarrow{\sigma}. \quad \exists\, T \sim T' : \overrightarrow{\tau}. \quad \frac{(\,S\,,\,M\,)}{(\,T\,,\,\star\,)} \wedge \frac{(\,S'\,,\,M'\,)}{(\,T'\,,\,\star\,)}$$

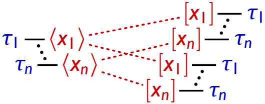
Theorem

$(\sim)$ Is a congruence and includes $\alpha\beta\eta$-equivalence.

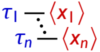
Theorem

Machine-equivalent FMC-terms form a Cartesian closed category

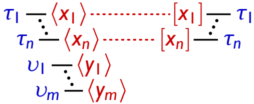
$$\delta : \overleftarrow{\tau} \longrightarrow \overleftarrow{\tau} \times \overleftarrow{\tau} \;=\; \langle \overleftarrow{x} \rangle . [\overrightarrow{x}] . [\overrightarrow{x}] : \overleftarrow{\tau} \Rightarrow \overrightarrow{\tau}\overrightarrow{\tau}$$



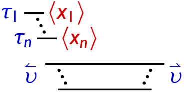
$$! : \overleftarrow{\tau} \longrightarrow 1 \;=\; \langle \overleftarrow{x} \rangle : \overleftarrow{\tau} \Rightarrow$$



$$\pi_1 : \overleftarrow{\tau} \times \overleftarrow{\upsilon} \longrightarrow \overleftarrow{\tau} \;=\; \langle \overleftarrow{x} \rangle . \langle \overleftarrow{y} \rangle . [\overrightarrow{x}] : \overleftarrow{\tau}\overleftarrow{\upsilon} \Rightarrow \overrightarrow{\tau}$$



$$\pi_2 : \overleftarrow{\tau} \times \overleftarrow{\upsilon} \longrightarrow \overleftarrow{\upsilon} \;=\; \langle \overleftarrow{x} \rangle : \overleftarrow{\tau}\overleftarrow{\upsilon} \Rightarrow \overrightarrow{\upsilon}$$



Theorem

Machine-equivalent FMC-terms form a Cartesian closed category

The exponent bifunctor:

$$\overleftarrow{\sigma} \to \overleftarrow{\tau} \quad = \quad \overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}$$

Its action on morphisms: for $M \colon \overleftarrow{\rho} \Rightarrow \overrightarrow{\sigma}$ and $N \colon \overleftarrow{\tau} \Rightarrow \overrightarrow{\upsilon}$

$$M \to N \;:\; (\overleftarrow{\sigma} \to \overleftarrow{\tau}) \longrightarrow (\overleftarrow{\rho} \to \overleftarrow{\upsilon}) \quad = \quad \langle f \rangle . [M . f . N] \colon (\overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}) \Rightarrow (\overleftarrow{\rho} \Rightarrow \overrightarrow{\upsilon})$$

The unit and counit of the adjunction $- \times \overleftarrow{\tau} \dashv \overleftarrow{\tau} \to -$

$$\epsilon \;:\; (\overleftarrow{\tau} \to \overleftarrow{\sigma}) \times \overleftarrow{\tau} \longrightarrow \overleftarrow{\sigma} \quad = \quad \langle f \rangle . f \;=_\eta\; \langle f \rangle . \langle \overleftarrow{x} \rangle . [\overrightarrow{x}] . f \quad : \; (\overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}) \overleftarrow{\sigma} \Rightarrow \overrightarrow{\tau}$$

$$\eta \;:\; \overleftarrow{\sigma} \longrightarrow (\overleftarrow{\tau} \to \overleftarrow{\sigma} \times \overleftarrow{\tau}) \quad = \quad \langle \overleftarrow{y} \rangle . [[\overrightarrow{y}]] \;=_\eta\; \langle \overleftarrow{y} \rangle . [\langle \overleftarrow{x} \rangle . [\overrightarrow{x}] . [\overrightarrow{y}]] \quad : \; \overleftarrow{\tau} \Rightarrow (\overleftarrow{\sigma} \Rightarrow \overrightarrow{\sigma} \overrightarrow{\tau})$$

Is this logic?