The Functional Machine Calculus III: Choice

(Early announcement)

Willem Heijltjes
University of Bath

MFPS 2024, Oxford

The Functional Machine Calculus (FMC)

A new model for combining λ**-calculus** with **computational effects**

Aims

- Confluence
- Types (strong normalization)
- Simplicity

Approach

- Operational semantics (stack machine) as primary
- Judicious choice of language constructs
- Decompose effect operators (rather than primitives)

[H 2022] [Barrett, H & McCusker 2023]

Previously: Locations and Sequencing

- Mutable store
- Input/output
- Probabilistic/non-deterministic sampling
- Imperative sequencing
- Strategies: CBV[1], computational metalanguage[2], CBPV[3]

This talk: Choice

- Exception handling (Exception monad: $TX = E + X$)
- Constants (E.g. Booleans: $\mathbb{B} = 1 + 1$)
- Data types (non-recursive)
- Iteration ($M : A \to A + B \quad \mapsto \quad \text{iter } M : A \to B$)

[1][Plotkin 1975] [2][Moggi 1991] [3][Levy 2003]

λ-Calculus: the machine

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

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

Stacks: $S ::= \varepsilon \mid S\, M$

States: $(S\ , M)$

Transitions:

$$\frac{(\ S\ ,\ [N].\, M\ )}{(\ S\, N\ ,\ M\ )}$$

$$\frac{(\ S\, N\ ,\ \langle x \rangle.\, M\ )}{(\ S\ ,\ \{N/x\}M\ )}$$

λ-Calculus: the machine

$$M, N \;::=\; \overbrace{x \;\mid\; [N].\,M \;\mid\; \langle x\rangle.\,M}^{\lambda\text{-calculus}}$$

Stacks: $S \;::=\; \varepsilon \mid S\,M$

States: $(S\;,M)$

Transitions:

$$\frac{(\,S\quad,\quad [N].\,M\,)}{(\,S\,N\quad,\quad M\,)}$$

$$\frac{(\,S\,N\quad,\quad \langle x\rangle.\,M\,)}{(\,S\quad,\quad \{N/x\}M\,)}$$

Locations

$$M, N ::= \overbrace{x \mid [N].\,M \mid \langle x\rangle.\,M}^{\lambda\text{-calculus}}$$

$$M, N ::= \underbrace{x \mid [N]a.\,M \mid a\langle x\rangle.\,M}_{\text{locations}}$$

Multiple stacks, named in a global set of **locations** $A = \{\lambda, a, b, c, \ldots\}$.

Push (application), **pop** (abstraction) parameterised in A — conservative by

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

Locations: the machine

$$\overbrace{M, N \;::=\; x \;\mid\; [N].\, M \;\mid\; \langle x\rangle.\, M}^{\lambda\text{-calculus}}$$

$$M, N \;::=\; \underbrace{x \;\mid\; [N]a.\, M \;\mid\; a\langle x\rangle.\, M}_{\text{locations}}$$

Stacks: $S \;::=\; \varepsilon \mid S\,M$

Memories: $S_A \;::=\; \{S_a \mid a \in A\}$

States: (S_A, M)

Transitions:

$$\frac{(\,S_A \cdot S_a \;,\; [N]a.\, M\,)}{(\,S_A \cdot (S\,N)_a \;,\; M\,)}$$

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

Effects

Input/output, **probabilities** as dedicated locations in, out, rnd

read: $\mathsf{in}\langle x\rangle.\,x$ print: $[N]\mathsf{out}.\,M$ random: $\mathsf{rnd}\langle x\rangle.\,M$

Store (mutable variables) as chosen locations a, b, c, . .

$$\begin{array}{llcl} \text{update:} & a := N\,;\,M & = & a\langle\_\rangle.\,[N]a.\,M \\ \text{lookup:} & !a & = & a\langle x\rangle.\,[x]a.\,x \qquad ({}^{1}) \end{array}$$

- **Confluence:** reduction equivalence includes algebraic store[2]
- **Typed termination:** Landin's Knot[3] cannot be typed

[1] Cf. Haskell MVars [Peyton Jones, Gordon & Finne 1996] [2] [Plotkin & Power 2002] [3] [Landin 1964]

Sequencing

$$\overbrace{\qquad\qquad\qquad\qquad}^{\lambda\text{-calculus}} \qquad \overbrace{\qquad\quad}^{\text{sequencing}}$$

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

$$M, N \;::=\; \underbrace{x \;\mid\; [N]a.\,M \;\mid\; a\langle x\rangle.\,M}_{\text{locations}}$$

Introduce imperative **skip** $\star$ and **sequence** $M\,;N$

identity and **composition** on the machine

Standard implementation: **continuation** stack where

$M\,;N$	pushes	N
$\star$	pops	

Cf. κ-calculus [Hasegawa 1995] [Power & Thielecke 1999], concatenative programming

Sequencing: the machine

$$M, N \ ::= \ \underbrace{x \mid [N].M \mid \langle x\rangle.M}_{\lambda\text{-calculus}} \mid \underbrace{\star \mid M;N}_{\text{sequencing}}$$

$$M, N \ ::= \ \underbrace{x \mid [N]a.M \mid a\langle x\rangle.M}_{\text{locations}}$$

Stacks: $S \ ::= \ \varepsilon \mid S\,M$ Memories: $S_A \ ::= \ \{S_a \mid a \in A\}$

States: (S_A, M, K) Continuation stacks: $K \ ::= \ \varepsilon \mid M\ K$

Transitions:

$$\frac{(\ S_A \cdot \ S_a \ , \ [N]a.M \ , \ K\ )}{(\ S_A \cdot (S\,N)_a \ , \ M \ , \ K\ )} \qquad \frac{(\ S_A \ , \ M;N \ , \ K\ )}{(\ S_A \ , \ M \ , \ N\ K\ )}$$

$$\frac{(\ S_A \cdot (S\,N)_a \ , \ a\langle x\rangle.M \ , \ K\ )}{(\ S_A \cdot \ S_a \ , \ \{N/x\}M \ , \ K\ )} \qquad \frac{(\ S_A \ , \ \star \ , \ N\ K\ )}{(\ S_A \ , \ N \ , \ K\ )}$$

Embedded calculi

$$M, N \;::=\; \overbrace{x \;\mid\; [N].\,M \;\mid\; \langle x\rangle.\,M}^{\lambda\text{-calculus}} \;\mid\; \overbrace{\star \;\mid\; M\,;N}^{\text{sequencing}}$$

$$M, N \;::=\; \underbrace{x \;\mid\; [N]a.\,M \;\mid\; a\langle x\rangle.\,M}_{\text{locations}}$$

CBV λ-calculus

$$x_v = x \qquad\qquad V_c = [V_v].\,\star$$

$$(\lambda x.M)_v = \langle x\rangle.\,M_c \qquad\qquad (M\,N)_c = N_c\;;\;M_c\;;\;\langle x\rangle.\,x$$

Computational metalanguage

$$\text{return } M = [M].\,\star \qquad\qquad \text{let } x = M \text{ in } N = M\;;\;\langle x\rangle.\,N$$

[Douence & Fradet 1998] [Power & Thielecke 1999]

Jumps and joins

$$M, N ::= \overbrace{\star \mid M ; N}^{\text{sequencing}}$$

Skip $\star$ signifies **successful termination**.

Jumps and joins

$$M, N ::= \underbrace{\overbrace{\star \mid M ; N}^{\text{sequencing}}}_{}$$

$$M, N ::= \underbrace{j \mid M ; j \to N}_{\text{choice}}$$

Skip $\star$ signifies **successful termination**.

Generalise to a set $\{\star, i, j, k, \ldots\}$ of **jumps** to include **modes of failure**.

Sequencing becomes a **join**, conditional on a given jump, conservative by

$$M ; N \;=\; M ;\, \star \to N$$

Jumps and joins

$$M, N \ ::= \ \overbrace{\star \ \mid \ M\,;N}^{\text{sequencing}}$$

States: (M, K) Continuation stacks: $K \ ::= \ \varepsilon \mid M \ K$

Transitions:

$$\frac{(\,M\,;N \ , \ K\,)}{(\,M \ , \ N \ K\,)}$$

$$\frac{(\,\star \ , \ N \ K\,)}{(\,N \ , \ K\,)}$$

Jumps and joins

$$M, N ::= \overbrace{\star \mid M ; N}^{\text{sequencing}}$$

$$M, N ::= \underbrace{j \mid M ; j \to N}_{\text{choice}}$$

States: (M, K) Continuation stacks: $K ::= \varepsilon \mid (j \to M)\, K$

Transitions:

$$\frac{(\,M ; j \to N\,,\; K\,)}{(\,M\,,\; (j \to N)\, K\,)}$$

$$\frac{(\,j\,,\; (j \to N)\, K\,)}{(\,N\,,\; K\,)}$$

$$\frac{(\,i\,,\; (j \to N)\, K\,)}{(\,i\,,\; K\,)} \; (i \neq j)$$

Jumps and joins

$$M, N \;::=\; \overbrace{\star \;\mid\; M\,;N}^{\text{sequencing}}$$

$$M, N \;::=\; \underbrace{j \;\mid\; M\,;j\rightarrow N}_{\text{choice}}$$

Exceptions are jumps:

$$\text{throw e} \;=\; \text{e}$$

$$\text{try } \{M\} \text{ catch e } \{N\} \;=\; M \;;\; \text{e}\rightarrow N$$

Jumps and joins

$$\underbrace{M, N ::= \overbrace{\star \mid M ; N}^{\text{sequencing}}}$$

$$M, N ::= \underbrace{j \mid M ; j \to N}_{\text{choice}}$$

Exceptions are jumps:

$$\begin{aligned} \text{throw e} &= e \\ \text{try } \{M\} \text{ catch e } \{N\} &= M \; ; \; e \to N \end{aligned}$$

Booleans are jumps:

$$\begin{aligned} \top, \bot &= \top, \bot \\ \text{if } B \text{ then } M \text{ else } N &= B ; \top \to M ; \bot \to N \\ &= (B ; \top \to M) ; \bot \to N \end{aligned}$$

Jumps and joins

$$M, N ::= \overbrace{\star \mid M ; N}^{\text{sequencing}}$$

$$M, N ::= \underbrace{j \mid M ; j \to N}_{\text{choice}}$$

Exceptions are jumps:

$$\text{throw e} = \text{e}$$

$$\text{try } \{M\} \text{ catch e } \{N\} = M \; ; \; \text{e} \to N$$

Booleans are jumps:

$$\top, \bot = \top, \bot$$

$$\begin{aligned} \text{if } B \text{ then } M \text{ else } N &= B ; \top \to M ; \bot \to N \\ &= (B ; \top \to M) ; \bot \to N \end{aligned}$$

Constants are jumps:

$$\text{case } M \text{ of } \{c_1 \to N_1, \ldots, c_n \to N_n\} = M ; c_1 \to N_1 \; ; \; \ldots \; ; c_n \to N_n$$

Iteration

$$M, N ::= \overbrace{\star \mid M\,;N}^{\text{sequencing}}$$

$$M, N ::= \underbrace{j \mid M\,; j \to N \mid M^j}_{\text{choice}}$$

A **loop** M^j repeats on j and exits on other jumps.

Iteration

$$M, N \;::=\; \underbrace{\star \;\mid\; M\,;N}^{\text{sequencing}}$$

$$M, N \;::=\; \underbrace{j \;\mid\; M\,;j\to N \;\mid\; M^j}_{\text{choice}}$$

States: (M, K) Continuation stacks: $K \;::=\; \varepsilon \mid (j\to M)\,K$

Transitions:

$$\frac{(\,M\,;j\to N\,,\;K\,)}{(\,M\,,\;(j\to N)\,K\,)}$$

$$\frac{(\,j\,,\;(j\to N)\,K\,)}{(\,N\,,\;K\,)}$$

$$\frac{(\,M^j\,,\;K\,)}{(\,M\,,\;(j\to M^j)\,K\,)} \qquad \frac{(\,i\,,\;(j\to N)\,K\,)}{(\,i\,,\;K\,)}\;(i\neq j)$$

Iteration

$$M, N \ ::= \ \overbrace{\star \ | \ M\,;N}^{\text{sequencing}}$$

$$M, N \ ::= \ \underbrace{j \ | \ M\,;j\!\to\! N \ | \ M^j}_{\text{choice}}$$

Do–while loops:

$$\text{do } M \text{ while } B \ = \ (M\,;B)^{\top}\,;\bot\!\to\!\star$$

Iteration

$$M, N ::= \overbrace{\star \mid M\,;N}^{\text{sequencing}}$$

$$M, N ::= \underbrace{j \mid M\,;j \to N \mid M^{j}}_{\text{choice}}$$

Do–while loops:

$$\text{do } M \text{ while } B \;=\; (M\,;B)^{\top}\,;\bot \to \star$$

While–do loops:

$$\text{while } B \text{ do } M \;=\; B\,;\top \to (M\,;B)^{\top}\,;\bot \to \star$$

$$\text{or } \;(B\,;\top \to M)^{\star}\,;\bot \to \star$$

Iteration

$$M, N \;::=\; \overbrace{\star \mid M\,;N}^{\text{sequencing}}$$

$$M, N \;::=\; \underbrace{j \mid M\,;j\to N \mid M^{j}}_{\text{choice}}$$

Do–while loops:

$$\text{do } M \text{ while } B \;=\; (M\,;B)^{\top}\,;\bot\to\star$$

While–do loops:

$$\text{while } B \text{ do } M \;=\; B\,;\top\to(M\,;B)^{\top}\,;\bot\to\star$$

$$\text{or } \; (B\,;\top\to M)^{\star}\,;\bot\to\star$$

Breaks are jumps:

$$\text{while true do } M \;=\; M^{\star}\,;\text{break}\to\star$$

Choice: the machine

$$\overbrace{M, N \;::=\; x \mid [N].\,M \mid \langle x\rangle.\,M}^{\lambda\text{-calculus}} \mid \overbrace{\star \mid M\,;N}^{\text{sequencing}}$$

$$M, N \;::=\; \underbrace{x \mid [N]a.\,M \mid a\langle x\rangle.\,M}_{\text{locations}} \mid \underbrace{j \mid M\,;j\!\to\! N \mid M^j}_{\text{choice}}$$

Stacks: $S \;::=\; \varepsilon \mid S\,M$ Memories: $S_A \;::=\; \{S_a \mid a \in A\}$

States: (S_A, M, K) Continuation stacks: $K \;::=\; \varepsilon \mid (j\!\to\! M)\,K$

Transitions:

$$\frac{(\,S_A \cdot S_a\,,\; [N]a.\,M\,,\; K\,)}{(\,S_A \cdot (S\,N)_a\,,\; M\,,\; K\,)} \qquad \frac{(\,S_A\,,\; M\,;j\!\to\! N\,,\; K\,)}{(\,S_A\,,\; M\,,\; (j\!\to\! N)\,K\,)}$$

$$\frac{(\,S_A \cdot (S\,N)_a\,,\; a\langle x\rangle.\,M\,,\; K\,)}{(\,S_A \cdot S_a\,,\; \{N/x\}M\,,\; K\,)} \qquad \frac{(\,S_A\,,\; j\,,\; (j\!\to\! N)\,K\,)}{(\,S_A\,,\; N\,,\; K\,)}$$

$$\frac{(\,S_A\,,\; M^j\,,\; K\,)}{(\,S_A\,,\; M\,,\; (j\!\to\! M^j)\,K\,)} \qquad \frac{(\,S_A\,,\; i\,,\; (j\!\to\! N)\,K\,)}{(\,S_A\,,\; i\,,\; K\,)}\;(i\neq j)$$

Data types

$$\overbrace{M, N ::= x \mid [N].M \mid \langle x\rangle.M}^{\lambda\text{-calculus}} \mid \overbrace{\star \mid M\,;N}^{\text{sequencing}}$$

$$M, N ::= \underbrace{x \mid [N]a.M \mid a\langle x\rangle.M}_{\text{locations}} \mid \underbrace{j \mid M\,;j\to N \mid M^j}_{\text{choice}}$$

Data constructors are jumps:

$$c\,M_1 \;\ldots\; M_n \;=\; [M_n]\ldots[M_1].\,c$$

Pattern-matching becomes unnecessary—arguments are passed on the stack

$$\text{case } M \text{ of } \{c_1\,\overline{x}_1 \to N_1, \ldots, c_n\,\overline{x}_n \to N_n\}$$

$$=$$

$$M \;;\; c_1 \to \langle \overline{x}_1 \rangle.\,N_1 \;;\; \ldots \;;\; c_n \to \langle \overline{x}_n \rangle.\,N_n$$

Example: factorial in a CBV language

$$\text{fac}\,x \quad = \quad c := x \;;\; a := 1 \;;\; \text{while}\; c > 1 \;\text{do}\; (\, a := a \times c \;;\; c := c - 1 \,) \;;\; a$$

$$\begin{aligned}
a := M &= M \;;\; \langle x \rangle .\, a\langle _ \rangle .\, [x]a \\
a &= a\langle x \rangle .\, [x]a .\, [x] \\
x &= [x] \\
M \times N &= M \,;\, N \,;\, \times \\
\text{while}\; M \;\text{do}\; N &= (M \,;\, \langle x \rangle .\, x \,;\, \top \to N)^{\star} \,;\, \bot \to \star \\
(f\, x_1 \ldots x_n = M) \;;\; N &= [\langle x_1 \rangle \ldots \langle x_n \rangle .\, M] .\, \langle f \rangle .\, N
\end{aligned}$$

Sequencing: types

$$M, N \ ::= \ \overbrace{x \ \mid \ [N].\,M \ \mid \ \langle x\rangle.\,M}^{\lambda\text{-calculus}} \ \mid \ \overbrace{\star \ \mid \ M\,;N}^{\text{sequencing}}$$

Types indicate the **input stack** and **return stack** on the machine

$$\sigma_1 \dots \sigma_n \ \Rightarrow \ \tau_1 \dots \tau_m$$

Semantics is given by the machine as a function on stacks

$$([\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!]) \rightarrow ([\![\tau_1]\!] \times \cdots \times [\![\tau_m]\!])$$

Sequencing: types

$$M, N \;::=\; \overbrace{x \;\mid\; [N].\,M \;\mid\; \langle x\rangle.\,M}^{\lambda\text{-calculus}} \;\mid\; \overbrace{\star \;\mid\; M\,;N}^{\text{sequencing}}$$

Types:	$\rho,\ \sigma,\ \tau \;::=\; \overline{\sigma} \Rightarrow \overline{\tau}$		$[\![\overline{\sigma}]\!] \to [\![\overline{\tau}]\!]$
Stack types:	$\overline{\tau} \;::=\; \tau_1 \dots \tau_n$		$[\![\tau_1]\!] \times \cdots \times [\![\tau_n]\!]$

$$S : \overline{\sigma}\ , \quad M : \overline{\sigma} \Rightarrow \overline{\tau} \qquad \Longrightarrow \qquad \exists T : \overline{\tau}. \quad \frac{(\,S\ ,\ M\ ,\ \varepsilon\,)}{(\,T\ ,\ \star\ ,\ \varepsilon\,)}$$

Category: (strict) CCC

Objects: type vectors $\overline{\tau}$

$$\text{Product:} \quad \overline{\sigma} \times \overline{\tau} = \overline{\sigma}\,\overline{\tau}$$

$$\text{Closure:} \quad \overline{\sigma} \to \overline{\tau} = \overline{\sigma} \Rightarrow \overline{\tau}$$

Morphisms: closed terms M

identity	$\star$	:	$\overline{\tau} \Rightarrow \overline{\tau}$
composition	$M\,;N$	:	$\overline{\rho} \Rightarrow \overline{\tau}$ for $M : \overline{\rho} \Rightarrow \overline{\sigma}, N : \overline{\sigma} \Rightarrow \overline{\tau}$
terminal	$\langle \overline{x} \rangle . \star$	:	$\overline{\tau} \Rightarrow \mathrm{I}$
diagonal	$\langle \overline{x} \rangle . [\overline{x}] . [\overline{x}] . \star$	:	$\overline{\tau} \Rightarrow \overline{\tau}\,\overline{\tau}$
eval	$\langle x \rangle . x$	:	$(\overline{\sigma} \Rightarrow \overline{\tau})\,\overline{\sigma} \Rightarrow \overline{\tau}$
eta	$\langle \overline{x} \rangle . [[\overline{x}] . \star] . \star$	:	$\overline{\sigma} \Rightarrow (\overline{\tau} \Rightarrow \overline{\tau}\,\overline{\sigma})$

Cf. [Lambek & Scott 1988]

Embedded calculi

CBN λ-calculus

$$\sigma_1 \to \ldots \to \sigma_n \to o \;=\; \sigma_1 \ldots \sigma_n \Rightarrow 1$$

CBV λ-calculus

$$\begin{aligned} x_v &= x & o_v &= 1 \Rightarrow 1 \\ (\lambda x.M)_v &= \langle x \rangle.\, M_c & (\sigma \to \tau)_v &= \sigma_v \Rightarrow \tau_v \\ V_c &= [V_v].\, \star & \tau_c &= 1 \Rightarrow \tau_v \\ (M\,N)_c &= N_c \;;\; M_c \;;\; \langle x \rangle.\, x \end{aligned}$$

Computational metalanguage

$$\begin{aligned} \text{return } M &= [M].\, \star & \sigma_1 \to \cdots \to \sigma_n \to T\tau &= \sigma_1 \ldots \sigma_n \Rightarrow \tau \\ \text{let } x = M \text{ in } N &= M \;;\; \langle x \rangle.\, N \end{aligned}$$

Locations: types

$$\overbrace{M, N ::= x \mid [N].M \mid \langle x\rangle.M}^{\lambda\text{-calculus}} \mid \overbrace{\star \mid M\,;N}^{\text{sequencing}}$$

$$M, N ::= \underbrace{x \mid [N]a.M \mid a\langle x\rangle.M}_{\text{locations}}$$

Types:	$\rho, \sigma, \tau ::= \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}$	$[\![\overline{\overline{\sigma}}]\!] \to [\![\overline{\overline{\tau}}]\!]$
Stack types:	$\overline{\tau} ::= \tau_1 \ldots \tau_n$	$[\![\tau_1]\!] \times \cdots \times [\![\tau_n]\!]$
Memory types:	$\overline{\overline{\tau}} ::= \{\overline{\tau}_a \mid a \in A\}$	$\prod_{a \in A} [\![\overline{\tau}_a]\!]$

$$S_A : \overline{\overline{\sigma}}\,,\ M : \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}} \quad\Longrightarrow\quad \exists T_A : \overline{\overline{\tau}}.\ \frac{(\,S_A\,,\ M\,,\ \varepsilon\,)}{(\,T_A\,,\ \star\,,\ \varepsilon\,)}$$

Store

Notation: memory types concatenate pointwise: $\overline{\overline{\sigma}}\,\overline{\overline{\tau}} \;=\; \{\overline{\sigma}_a\,\overline{\tau}_a \mid a \in A\}$
singleton memory types: $a(\overline{\tau})$

$$\text{update} \quad \langle x\rangle.\,a\langle\_\rangle.\,[x]a \;:\; \tau\, a(\tau) \Rightarrow a(\tau)$$

$$\text{lookup} \quad a\langle x\rangle.\,[x]a.\,[x] \;:\; a(\tau) \Rightarrow a(\tau)\,\tau$$

$$\text{fac}\; x \;=\; c := x \;;\; a := 1 \;;\; \text{while}\; c > 1 \;\text{do}\; (\, a := a \times c \;;\; c := c - 1 \,) \;;\; a$$

$$> \;:\; \mathbb{Z}\,\mathbb{Z} \Rightarrow \mathbb{B}$$

$$c > 1 \;=\; c\langle x\rangle.\,[x]c.\,[x]\,;[1]\,;> \;:\; c(\mathbb{Z}) \Rightarrow c(\mathbb{Z})\,\mathbb{B}$$

$$c := c - 1 \;=\; c\langle x\rangle.\,[x]c.\,[x]\,;[1]\,;-\,;\langle x\rangle.\,c\langle\_\rangle.\,[x]c \;:\; c(\mathbb{Z}) \Rightarrow c(\mathbb{Z})$$

$$a := a \times c \;;\; c := c - 1 \;:\; a(\mathbb{Z})\,c(\mathbb{Z}) \Rightarrow a(\mathbb{Z})\,c(\mathbb{Z})$$

Choice: types

$$\overbrace{M, N ::= x \mid [N].M \mid \langle x\rangle.M}^{\lambda\text{-calculus}} \mid \overbrace{\star \mid M;N}^{\text{sequencing}}$$

$$\underbrace{M, N ::= x \mid [N]a.M \mid a\langle x\rangle.M}_{\text{locations}} \mid \underbrace{j \mid M;j\to N \mid M^j}_{\text{choice}}$$

Types:	$\rho, \sigma, \tau ::= \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_J$	$[\![\overline{\overline{\sigma}}]\!] \to [\![\overline{\overline{\tau}}_J]\!]$
Stack types:	$\overline{\tau} ::= \tau_1 \dots \tau_n$	$[\![\tau_1]\!] \times \cdots \times [\![\tau_n]\!]$
Memory types:	$\overline{\overline{\tau}} ::= \{\overline{\tau}_a \mid a \in A\}$	$\prod_{a\in A} [\![\overline{\tau}_a]\!]$
Choice types:	$\overline{\overline{\tau}}_J ::= \{\overline{\overline{\tau}}_j \mid j \in J\}$	$\sum_{j\in J} [\![\overline{\overline{\tau}}_j]\!]$

$$S_A : \overline{\overline{\sigma}}\ ,\ M : \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_J \quad \Longrightarrow \quad \exists j \in J.\ \exists T_A : \overline{\overline{\tau}}_j.\ \ \frac{(\ S_A\ ,\ M\ ,\ \varepsilon\ )}{(\ T_A\ ,\ j\ ,\ \varepsilon\ )}$$

Choice: types

Notation: sum to compose choice types $\overline{\overline{\sigma}}_I + \overline{\overline{\tau}}_J$ ($I \cap J = \varnothing$)

$$\frac{}{i : \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\sigma}}_i + \overline{\overline{\tau}}_J} \qquad \frac{M : \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_J + \overline{\overline{\rho}}_i \qquad N : \overline{\overline{\rho}} \Rightarrow \overline{\overline{\tau}}_J}{M \,;\, i \to N : \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_J} \qquad \frac{M : \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\sigma}}_i + \overline{\overline{\tau}}_J}{M^i : \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_J}$$

Typing factorial:

$$\text{fac}\, x \;=\; c := x \;;\; a := 1 \;;\; \text{while } c > 1 \text{ do } (\, a := a \times c \;;\; c := c - 1 \,) \;;\; a$$

$$\top, \bot \;:\; \overline{\overline{\tau}} \Rightarrow \overline{\overline{\tau}}_\top + \overline{\overline{\tau}}_\bot$$

$$B : \overline{\overline{\tau}} \Rightarrow \overline{\overline{\tau}}_\top + \overline{\overline{\tau}}_\bot \,,\; M : \overline{\overline{\tau}} \Rightarrow \overline{\overline{\tau}}_\star \;\Longrightarrow\; B \,;\, \top \to M \;:\; \overline{\overline{\tau}} \Rightarrow \overline{\overline{\tau}}_\bot + \overline{\overline{\tau}}_\star$$

$$(B \,;\, \top \to M)^\star \;:\; \overline{\overline{\tau}} \Rightarrow \overline{\overline{\tau}}_\bot$$

$$\text{while } B \text{ do } M \;=\; (B \,;\, \top \to M)^\star \,;\, \bot \to \star \;:\; \overline{\overline{\tau}} \Rightarrow \overline{\overline{\tau}}_\star$$

$$\text{while } c > 1 \text{ do } (\, a := a \times c \;;\; c := c - 1 \,) \;:\; a(\mathbb{Z})\, c(\mathbb{Z}) \Rightarrow (a(\mathbb{Z})\, c(\mathbb{Z}))_\star$$

$$\text{fac} \;:\; \mathbb{Z}\, a(\mathbb{Z})\, c(\mathbb{Z}) \Rightarrow (\mathbb{Z}\, a(\mathbb{Z})\, c(\mathbb{Z}))_\star$$

Type system

$$\frac{}{\Gamma, x\colon \tau \vdash x\colon \tau} \qquad \frac{}{\Gamma \vdash j\colon \mathsf{I} \Rightarrow \mathsf{I}_j}$$

$$\frac{\Gamma \vdash M\colon \overline{\overline{\rho}} \Rightarrow \overline{\overline{\tau}}_J}{\Gamma \vdash M\colon \overline{\overline{\rho}}\,\overline{\overline{\sigma}} \Rightarrow (\overline{\overline{\sigma}}\,\overline{\overline{\tau}})_J} \qquad \frac{\Gamma \vdash M\colon \overline{\overline{\rho}} \Rightarrow \overline{\overline{\tau}}_J}{\Gamma \vdash M\colon \overline{\overline{\rho}} \Rightarrow \overline{\overline{\sigma}}_I + \overline{\overline{\tau}}_J}$$

$$\frac{\Gamma \vdash N\colon \rho \quad \Gamma \vdash M\colon a(\rho)\,\overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_J}{\Gamma \vdash [N]a.\,M\colon \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_J} \qquad \frac{\Gamma \vdash N\colon \overline{\overline{\rho}} \Rightarrow \overline{\overline{\tau}}_I + \overline{\overline{\sigma}}_j \quad \Gamma \vdash M\colon \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_I}{\Gamma \vdash N\,;j \rightarrow M\colon \overline{\overline{\rho}} \Rightarrow \overline{\overline{\tau}}_I}$$

$$\frac{\Gamma, x\colon \rho \vdash M\colon \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_J}{\Gamma \vdash a\langle x\rangle.\,M\colon a(\rho)\,\overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_J} \qquad \frac{\Gamma \vdash M\colon \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_I + \overline{\overline{\sigma}}_j}{\Gamma \vdash M^{j}\colon \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_I}$$

Reduction

$$M, N \ ::= \ \overbrace{x \ \mid \ [N].\,M \ \mid \ \langle x\rangle.\,M}^{\lambda\text{-calculus}}$$

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

Reduction

$$M, N \;::=\; \overbrace{x \;\mid\; [N].\,M \;\mid\; \langle x\rangle.\,M}^{\lambda\text{-calculus}}$$

$$M, N \;::=\; \underbrace{x \;\mid\; [N]a.\,M \;\mid\; a\langle x\rangle.\,M}_{\text{locations}}$$

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

$$[N]b.\,a\langle x\rangle.\,M \;\longrightarrow\; a\langle x\rangle.\,[N]b.\,M \qquad (a \neq b, x \notin \mathsf{fv}(N))$$

Reduction

$$M, N \;::=\; \overbrace{x \;\mid\; [N].\,M \;\mid\; \langle x \rangle.\,M}^{\lambda\text{-calculus}} \;\mid\; \overbrace{\star \;\mid\; M\,;N}^{\text{sequencing}}$$

$$M, N \;::=\; \underbrace{x \;\mid\; [N]a.\,M \;\mid\; a\langle x \rangle.\,M}_{\text{locations}}$$

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

$$[N]b.\,a\langle x \rangle.\,M \;\longrightarrow\; a\langle x \rangle.\,[N]b.\,M \qquad (a \neq b,\, x \notin \mathsf{fv}(N))$$

$$\star\,;P \;\longrightarrow\; P$$

$$([N].\,M)\,;P \;\longrightarrow\; [N].\,(M\,;P)$$

$$(\langle x \rangle.\,M)\,;P \;\longrightarrow\; \langle x \rangle.\,(M\,;P) \qquad (x \notin \mathsf{fv}(P))$$

$$(M\,;N)\,;P \;\longrightarrow\; M\,;(N\,;P)$$

Reduction

$$M, N ::= \overbrace{x \mid [N].M \mid \langle x\rangle.M}^{\lambda\text{-calculus}} \mid \overbrace{\star \mid M\,;N}^{\text{sequencing}}$$

$$M, N ::= \underbrace{x \mid [N]a.M \mid a\langle x\rangle.M}_{\text{locations}} \mid \underbrace{j \mid M\,;j{\to}N \mid M^j}_{\text{choice}}$$

$$
\begin{array}{rcll}
[N]a.\,a\langle x\rangle.\,M & \longrightarrow & \{N/x\}M & \\
[N]b.\,a\langle x\rangle.\,M & \longrightarrow & a\langle x\rangle.\,[N]b.\,M & (a \neq b, x \notin \mathrm{fv}(N)) \\
j\,;j{\to}P & \longrightarrow & P & \\
i\,;j{\to}P & \longrightarrow & i & (i \neq j) \\
([N].\,M)\,;j{\to}P & \longrightarrow & [N].\,(M\,;j{\to}P) & \\
(\langle x\rangle.\,M)\,;j{\to}P & \longrightarrow & \langle x\rangle.\,(M\,;j{\to}P) & (x \notin \mathrm{fv}(P)) \\
(M\,;N)\,;P & \longrightarrow & M\,;(N\,;P) & \\
M^j & \longrightarrow & M\,;j{\to}M^j &
\end{array}
$$

Proofs (without choice)

Machine termination:

- Use the meaning of types as **reducibility predicates**

$$\mathrm{RED}(\overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}) \;=\; \{M \mid \forall S_A \in \mathrm{RED}(\overline{\overline{\sigma}}).\ \exists T_A \in \mathrm{RED}(\overline{\overline{\tau}}).\ \frac{(\ S_A\ ,\ M\ ,\ \varepsilon\ )}{(\ T_A\ ,\ \star\ ,\ \varepsilon\ )}\}$$

- Proof by structural induction on typing derivations

Strong normalization:

- Machine termination gives a run with a certain length
  (A suitable stack exists because all types are inhabited)
- Beta-reduction shortens the run
- Computing this directly gives a Gandy-style SN proof

Confluence:

- By standard parallel reduction

Overview

Established for locations and sequencing; expected for choice:

- Confluence
- Typed machine termination, strong normalization (without loops)

Arguments for simplicity

- A complete typed programming language in six constructors
- Seamless integration of λ-calculus, sequencing, effects
- Intuitive abstract machine, using only stacks
- Semantics in sums, products, and function spaces

Implementation

- Normalize (supercompile) except loop-unrolling
- Lambda-lift to supercombinators
- Run