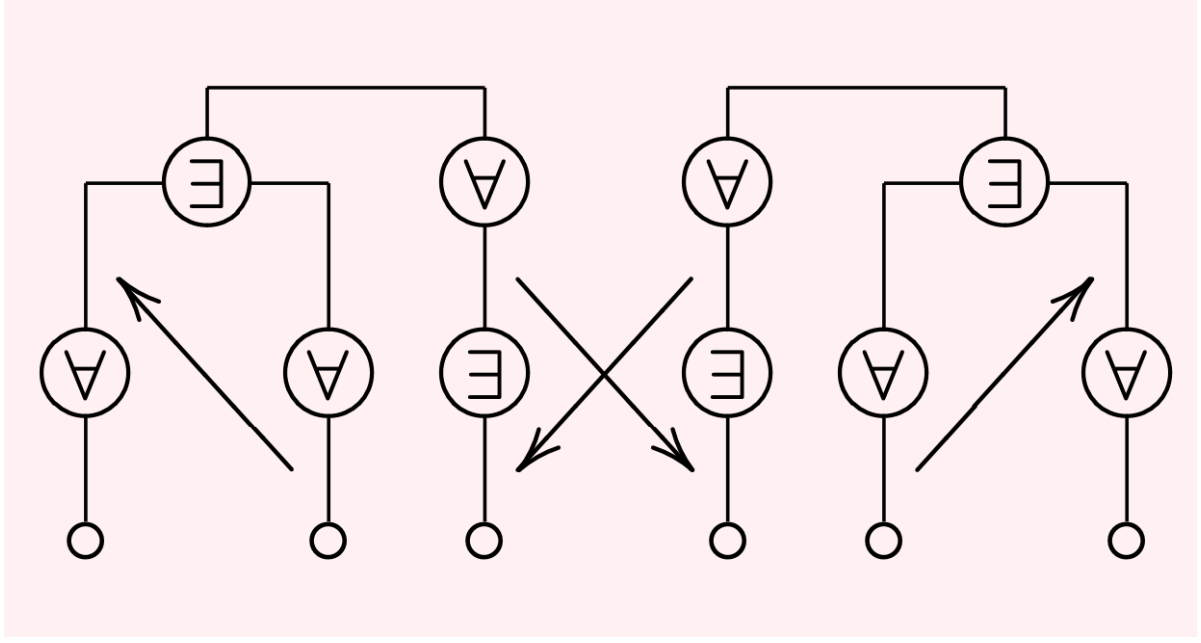
Observations in Semantics via the Functional Machine Calculus

Willem Heijltjes
University of Bath

MFPS, 3 June 2026, Ljubljana
Special Session for Alex Simpson's birthday

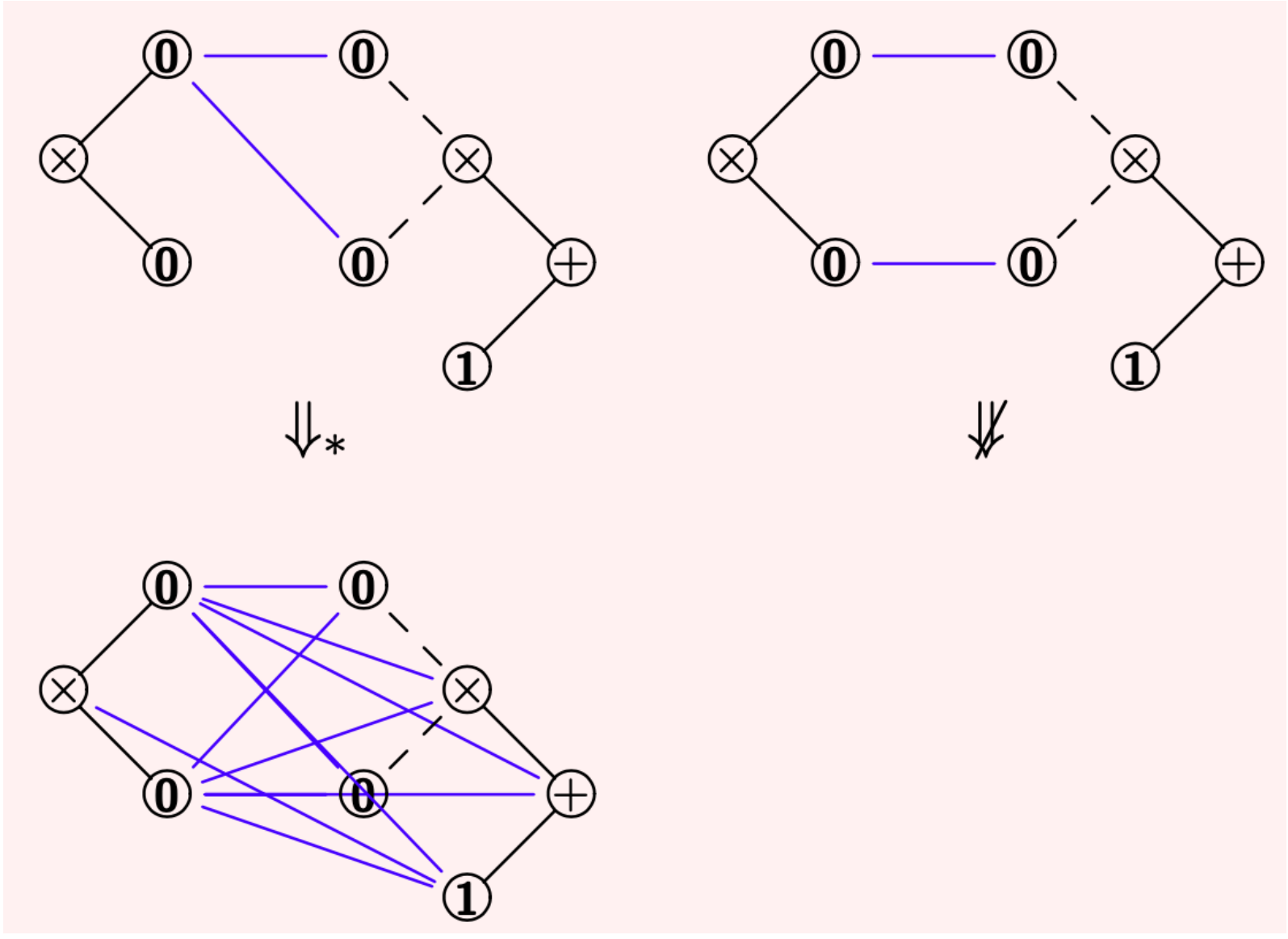
Proof forests

2008



Fascinating!!!

— Alex Simpson


0
0
0
0
1
⇓*
0
0
0
0
1
⇓
0
0
0
0
1

Lessons from Alex

Look at interesting things

Lessons from Alex

Look at interesting things

We need syntax (proofs!) and semantics

Lessons from Alex (and being in Edinburgh)

Look at interesting things

We need syntax (proofs!) and semantics

(Everybody else is doing programming language semantics)

The Functional Machine Calculus

$$M, N ::= \overbrace{x \mid [N]a.\,M \mid a\langle x\rangle.\,M}^{\lambda\text{-calculus + effects}} \mid \overbrace{i \mid M\,;i{\to}N \mid M^i}^{\text{branching + iteration}}$$

λ-Calculus + effects

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

Krivine Abstract Machine[1] with multiple stacks λ, a, b, c, $\ldots \in A$

- $[N]a.\, M$ pushes N to stack a, continues as M
- $a\langle x\rangle.\, M$ pops (say) N from stack a, continues as $\{N/x\}M$

Types indicate the expected input stacks:

$$\begin{array}{lll} \tau ::= \overline{\overline{\tau}} \Rightarrow \varepsilon & = & \overline{\overline{\tau}} \to \mathsf{I} \\ \overline{\tau} ::= \tau_1 \ldots \tau_n & = & \tau_1 \times \cdots \times \tau_n \\ \overline{\overline{\tau}} ::= \{\overline{\tau}^a \mid a \in A\} & = & \prod_{a \in A} \overline{\tau}^a \end{array}$$

[1] [Krivine 2007]

Call–by–name λ-calculus:

$$\begin{aligned} M\,N &= [N]\lambda.\,M \\ \lambda x.M &= \lambda\langle x\rangle.\,M \end{aligned}$$

Higher-order store:

$$\begin{aligned} a := M\ ;\ N &= a\langle\_\rangle.\,[M]a.\,N \\ !a &= a\langle x\rangle.\,[x]a.\,x \end{aligned}$$

Input/output:

$$\begin{aligned} \mathsf{read} &= \mathsf{stdin}\langle x\rangle.\,x \\ \mathsf{print}\ M\ ;\ N &= [M]\mathsf{stdout}.\,N \end{aligned}$$

Probabilistic choice:

$$\mathsf{coin} = \mathsf{rnd}\langle x\rangle.\,x$$

Branching + iteration

Computation with branches $\star$, i, j, k, $\ldots$

Terms and types:

$$M, N ::= i \mid M\,;i{\to}N \mid M^i$$

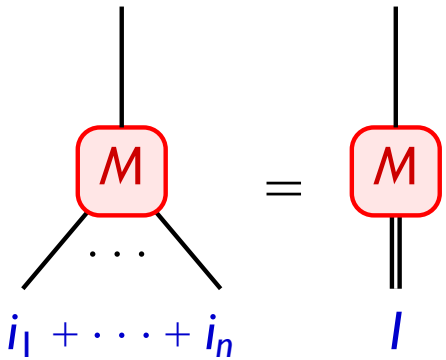
$$I, J ::= i_1 + \cdots + i_n$$

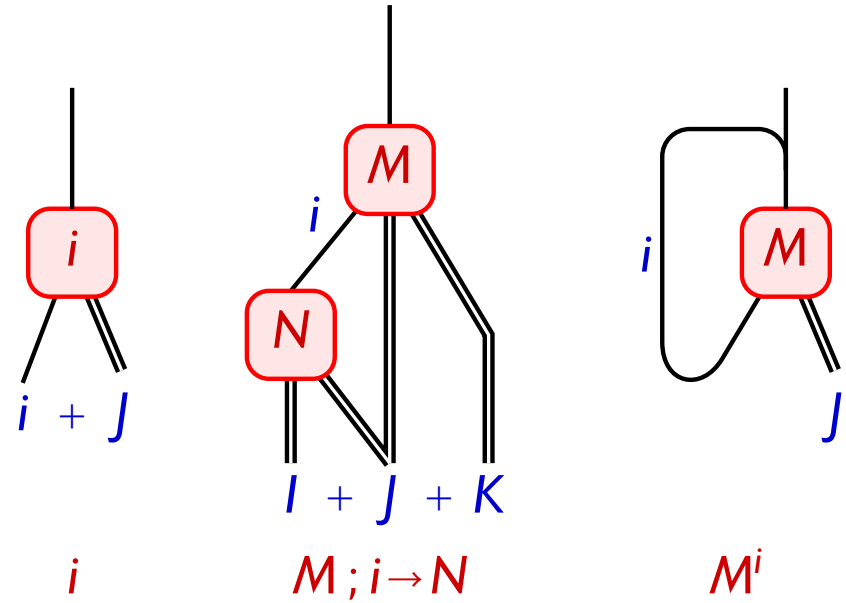
Computation:

$$i\,;i{\to}M \longrightarrow M$$

$$i\,;j{\to}M \longrightarrow i \qquad (i \neq j)$$

$$M^i \longrightarrow M\,;i{\to}M^i$$





Implemented with a stack K of **conditional continuations** $(i \to M)$

$$\frac{(\ M\ ;\ i \to N\ ,\qquad\qquad K\ )}{(\ M\qquad\qquad ,\quad (i \to N)\ K\ )} \qquad \frac{(\ i\quad ,\ (i \to N)\ K\ )}{(\ N\ ,\qquad\qquad K\ )}$$

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

See also [Benton & Kennedy 2001]

Sequencing:

$$\begin{aligned} \text{skip} &= \star \\ M; N &= M\,;\, \star \to N \end{aligned}$$

Exception handling:

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

Constants:

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

Loops:

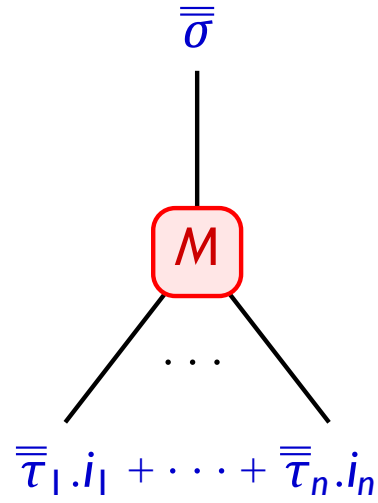
$$\text{while } B \text{ do } M = (B\,;\, \top \to M)^{\star}\,;\, \bot \to \star$$

The Functional Machine Calculus

$$M, N ::= \overbrace{x \mid [N]a.M \mid a\langle x\rangle.M}^{\lambda\text{-calculus + effects}} \mid \overbrace{i \mid M\,;i{\to}N \mid M^i}^{\text{branching + iteration}}$$

Types indicate the **input** and **output** stacks for each branch:

$$\begin{aligned}
\sigma, \tau &::= \overline{\overline{\sigma}} \Rightarrow \overline{\overline{\tau}}_I &&= \overline{\overline{\tau}} \to \overline{\overline{\tau}}_I \\
\overline{\tau} &::= \tau_1 \ldots \tau_n &&= \tau_1 \times \cdots \times \tau_n \\
\overline{\overline{\tau}} &::= \{\overline{\tau}^a \mid a \in A\} &&= \textstyle\prod_{a\in A} \overline{\tau}^a \\
\overline{\overline{\tau}}_I &::= \{\overline{\overline{\tau}}_i \mid i \in I\} &&= \textstyle\sum_{i\in I} \overline{\overline{\tau}}_i
\end{aligned}$$



Observation 1

A duality between exceptions and states

[Dumas, Duval, Fousse & Reynaud 2012]

State	Exceptions
Indexed product	Indexed sum

State	**Exceptions**
Indexed product	Indexed sum
$[N]a.\,M$	$M\,;i{\to}N$

State	**Exceptions**
Indexed product	Indexed sum
$[N]a.\,M$	$M\,;i{\to}N$
$B \overset{[N]a}{\Longrightarrow} B \times A_a \overset{M}{\Longrightarrow} C$	$C \overset{M}{\Longrightarrow} A_i + B \overset{i\to N}{\Longrightarrow} B$

State	**Exceptions**
Indexed product	Indexed sum
$[N]a.\,M$	$M\,;i\to N$
$B \overset{[N]a}{\Longrightarrow} B \times A_a \overset{M}{\Longrightarrow} C$	$C \overset{M}{\Longrightarrow} A_i + B \overset{i\to N}{\Longrightarrow} B$
	i
	$A \Longrightarrow A_i + B$

State	**Exceptions**
Indexed product	Indexed sum
$[N]a.\,M$	$M\,;i{\to}N$
$B \overset{[N]a}{\Longrightarrow} B \times A_a \overset{M}{\Longrightarrow} C$	$C \overset{M}{\Longrightarrow} A_i + B \overset{i\to N}{\Longrightarrow} B$
$a\langle\_\rangle.\star$	i
$A_a \times B \Longrightarrow B$	$A \Longrightarrow A_i + B$

State	**Exceptions**
Indexed product	Indexed sum
$[N]a.\,M$	$M\,;i\to N$
$B \overset{[N]a}{\Longrightarrow} B \times A_a \overset{M}{\Longrightarrow} C$	$C \overset{M}{\Longrightarrow} A_i + B \overset{i\to N}{\Longrightarrow} B$
$a\langle\_\rangle.\star$	i
$A_a \times B \Longrightarrow B$	$A \Longrightarrow A_i + B$
	$M^i \longrightarrow M\,;i\to M^i$
	$A \overset{M}{\Longrightarrow} A_i + B \overset{i\to M^i}{\Longrightarrow} B$

State	**Exceptions**
Indexed product	Indexed sum
$[N]a.\,M$	$M\,;i\to N$
$B \overset{[N]a}{\Longrightarrow} B \times A_a \overset{M}{\Longrightarrow} C$	$C \overset{M}{\Longrightarrow} A_i + B \overset{i\to N}{\Longrightarrow} B$
$a\langle\_\rangle.\star$	i
$A_a \times B \Longrightarrow B$	$A \Longrightarrow A_i + B$
$Y_a\,M \longrightarrow [Y_a\,M]a.\,M$	$M^i \longrightarrow M\,;i\to M^i$
	$A \overset{M}{\Longrightarrow} A_i + B \overset{i\to M^i}{\Longrightarrow} B$

State	**Exceptions**
Indexed product	Indexed sum
$[N]a.\,M$	$M\,;i\to N$
$B \overset{[N]a}{\Longrightarrow} B \times A_a \overset{M}{\Longrightarrow} C$	$C \overset{M}{\Longrightarrow} A_i + B \overset{i\to N}{\Longrightarrow} B$
$a\langle\_\rangle.\star$	i
$A_a \times B \Longrightarrow B$	$A \Longrightarrow A_i + B$
$Y_a\,M \longrightarrow [Y_a\,M]a.\,M$	$M^i \longrightarrow M\,;i\to M^i$
$B \overset{[Y_a\,M]a}{\Longrightarrow} B \times (B \Rightarrow C)_a \overset{M}{\Longrightarrow} C$	$A \overset{M}{\Longrightarrow} A_i + B \overset{i\to M^i}{\Longrightarrow} B$

State	**Exceptions**	**Contexts**
Indexed product	Indexed sum	Indexed product
$[N]a.\,M$	$M\,;i\to N$	
$B \overset{[N]a}{\Longrightarrow} B \times A_a \overset{M}{\Longrightarrow} C$	$C \overset{M}{\Longrightarrow} A_i + B \overset{i\to N}{\Longrightarrow} B$	
$a\langle\_\rangle.\star$	i	x
$A_a \times B \Longrightarrow B$	$A \Longrightarrow A_i + B$	$\Gamma \times A_x \longrightarrow A$
$Y_a\,M \rightarrowtriangle [Y_a\,M]a.\,M$	$M^i \rightarrowtriangle M\,;i\to M^i$	
$B \overset{[Y_a\,M]a}{\Longrightarrow} B \times (B \Rightarrow C)_a \overset{M}{\Longrightarrow} C$	$A \overset{M}{\Longrightarrow} A_i + B \overset{i\to M^i}{\Longrightarrow} B$	

State	**Exceptions**	**Contexts**
Indexed product	Indexed sum	Indexed product
$[N]a.\,M$	$M\,;i{\to}N$	$\{N/x\}M$
$B \overset{[N]a}{\Longrightarrow} B \times A_a \overset{M}{\Longrightarrow} C$	$C \overset{M}{\Longrightarrow} A_i + B \overset{i\to N}{\Longrightarrow} B$	$\Gamma \overset{\{N/x\}}{\longrightarrow} \Gamma \times A_x \overset{M}{\longrightarrow} C$
$a\langle\_\rangle.\star$	i	x
$A_a \times B \Longrightarrow B$	$A \Longrightarrow A_i + B$	$\Gamma \times A_x \longrightarrow A$
$Y_a\,M \rightarrowtriangle [Y_a\,M]a.\,M$	$M^i \rightarrowtriangle M\,;i{\to}M^i$	
$B \overset{[Y_a\,M]a}{\Longrightarrow} B \times (B \Rightarrow C)_a \overset{M}{\Longrightarrow} C$	$A \overset{M}{\Longrightarrow} A_i + B \overset{i\to M^i}{\Longrightarrow} B$	

State	**Exceptions**	**Contexts**
Indexed product	Indexed sum	Indexed product
$[N]a.\,M$	$M\,;i{\to}N$	$\{N/x\}M$
$B \overset{[N]a}{\Longrightarrow} B \times A_a \overset{M}{\Longrightarrow} C$	$C \overset{M}{\Longrightarrow} A_i + B \overset{i\to N}{\Longrightarrow} B$	$\Gamma \overset{\{N/x\}}{\longrightarrow} \Gamma \times A_x \overset{M}{\longrightarrow} C$
$a\langle\_\rangle.\star$	i	x
$A_a \times B \Longrightarrow B$	$A \Longrightarrow A_i + B$	$\Gamma \times A_x \longrightarrow A$
$Y_a\,M \longrightarrow [Y_a\,M]a.\,M$	$M^i \longrightarrow M\,;i{\to}M^i$	$\mu x.\,M \longrightarrow \{\mu x.\,M/x\}M$
$B \overset{[Y_a M]a}{\Longrightarrow} B \times (B \Rightarrow C)_a \overset{M}{\Longrightarrow} C$	$A \overset{M}{\Longrightarrow} A_i + B \overset{i\to M^i}{\Longrightarrow} B$	$\Gamma \overset{\{\mu x.M/x\}}{\longrightarrow} \Gamma \times A_x \overset{M}{\longrightarrow} A$

	Exceptions	**Contexts**
	Indexed sum	Indexed product
	$M\,;i\to N$	$\{N/x\}M$
	$C \overset{M}{\Longrightarrow} A_i + B \overset{i\to N}{\Longrightarrow} B$	$\Gamma \overset{\{N/x\}}{\longrightarrow} \Gamma \times A_x \overset{M}{\longrightarrow} C$
	i	x
	$A \Longrightarrow A_i + B$	$\Gamma \times A_x \longrightarrow A$
	$M^i \longrightarrow M\,;i\to M^i$	$\mu x.\,M \longrightarrow \{\mu x.\,M/x\}M$
	$A \overset{M}{\Longrightarrow} A_i + B \overset{i\to M^i}{\Longrightarrow} B$	$\Gamma \overset{\{\mu x.\,M/x\}}{\longrightarrow} \Gamma \times A_x \overset{M}{\longrightarrow} A$

Values: $\prod_{x \in X} A_x \longrightarrow B$

Computations: $A \underset{\Gamma}{\Longrightarrow} \sum_{i \in I} B_i$

Observation 2

Continuations are the "mother of all monads" [Filinski 1994]

vs

Typed Exceptions and Continuations Cannot Macro-Express Each Other [Riecke & Thielecke 1999]

Continuation encodings

Sequential computation: one branch $\star$, single continuation variable r:

$$\begin{array}{rcl} \lfloor \star \rfloor & = & r \\ \lfloor M\,;N \rfloor & = & (\lambda r.\lfloor M \rfloor)\,\lfloor N \rfloor \;\; \longrightarrow \;\; \{\lfloor N \rfloor / r\}\lfloor M \rfloor \end{array}$$

Continuation encodings

Sequential computation: one branch $\star$, single continuation variable r:

$$
\begin{array}{rcl}
\lfloor \star \rfloor & = & r \\
\lfloor M\,;N \rfloor & = & (\lambda r.\lfloor M \rfloor)\ \lfloor N \rfloor \quad \rightarrowtriangle \quad \{\lfloor N \rfloor / r\} \lfloor M \rfloor \\
\lfloor x \rfloor & = & x\, r \\
\lfloor \langle x \rangle .\, M \rfloor & = & \lambda x.\lfloor M \rfloor \\
\lfloor [N].\, M \rfloor & = & \lfloor M \rfloor\ (\lambda r.\lfloor N \rfloor)
\end{array}
$$

Continuation encodings

Sequential computation: one branch $\star$, single continuation variable r:

$$\begin{array}{rcl} \lfloor \star \rfloor & = & r \\ \lfloor M\,;N \rfloor & = & (\lambda r.\lfloor M \rfloor)\ \lfloor N \rfloor \quad \rightarrowtriangle \quad \{\lfloor N \rfloor / r\} \lfloor M \rfloor \\ \lfloor x \rfloor & = & x\,r \\ \lfloor \langle x \rangle .\, M \rfloor & = & \lambda x.\lfloor M \rfloor \\ \lfloor [N].\, M \rfloor & = & \lfloor M \rfloor\ (\lambda r.\lfloor N \rfloor) \end{array}$$

Multiple fixed branches i, j as continuation variables i, j:

$$\begin{array}{rcl} \lfloor i \rfloor & = & i \\ \lfloor M\,; i \rightarrow N \rfloor & = & (\lambda i.\lfloor M \rfloor)\ \lfloor N \rfloor \quad \rightarrowtriangle \quad \{\lfloor N \rfloor / i\} \lfloor M \rfloor \end{array}$$

Continuation encodings

Sequential computation: one branch $\star$, single continuation variable r:

$$\begin{array}{rcl} \lfloor \star \rfloor & = & r \\ \lfloor M\,;N \rfloor & = & (\lambda r.\lfloor M \rfloor)\ \lfloor N \rfloor \quad \rightarrow \quad \{\lfloor N \rfloor / r\} \lfloor M \rfloor \\ \lfloor x \rfloor & = & x\,r \\ \lfloor \langle x \rangle .\, M \rfloor & = & \lambda x.\lfloor M \rfloor \\ \lfloor [N].\, M \rfloor & = & \lfloor M \rfloor\ (\lambda r.\lfloor N \rfloor) \end{array}$$

Multiple fixed branches i, j as continuation variables i, j:

$$\begin{array}{rcl} \lfloor i \rfloor & = & i \\ \lfloor M\,; i \rightarrow N \rfloor & = & (\lambda i.\lfloor M \rfloor)\ \lfloor N \rfloor \quad \rightarrow \quad \{\lfloor N \rfloor / i\} \lfloor M \rfloor \\ \lfloor x \rfloor & = & x\,i\,j \\ \lfloor \langle x \rangle .\, M \rfloor & = & \lambda x.\lfloor M \rfloor \\ \lfloor [N].\, M \rfloor & = & \lfloor M \rfloor\ (\lambda i j.\lfloor N \rfloor) \end{array}$$

We need to know the possible branches!

Continuation encodings of sequencing/branching take

$$\textbf{dynamic}\ \ M\,;i{\to}N\ \ \text{to}\ \ \textbf{static}\ \ \{N/i\}M$$

Branches must be known statically; **exactly** what types provide.

$$M\colon I \qquad\qquad M\colon \overline{\overline{\sigma}}\Rightarrow\overline{\overline{\tau}}_I$$

Essentially, it is then the higher-order encoding of sums:

$$A+B \;\;\mapsto\;\; \forall R.(A\to R)\to(B\to R)\to R$$

Observation 3

Consider the branching fragment with a probabilistic generator,

$$M\,;N \;::=\; i \;\mid\; M\,;i{\rightarrow}N \;\mid\; M^{i} \;\mid\; \mathrm{rnd}\langle x\rangle.\,x$$

where types are probablistic sums (i.e. sub-distributions):

$$M\colon i_1 + \ldots + i_n \quad\longmapsto\quad M : p_1 i_1 \oplus \cdots \oplus p_n i_n$$

Observation 3

Consider the branching fragment with a probabilistic generator,

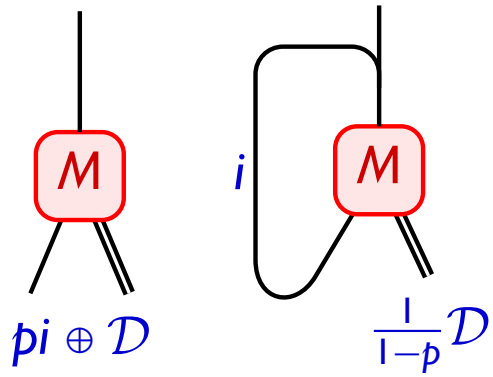
$$M\,;N \;::=\; i \;\mid\; M\,;i{\to}N \;\mid\; M^i \;\mid\; \mathrm{rnd}\langle x\rangle.\,x$$

where types are probablistic sums (i.e. sub-distributions):

$$M\colon i_1 + \ldots + i_n \quad\longmapsto\quad M : p_1 i_1 \oplus \cdots \oplus p_n i_n$$

Typing for loops adds **renormalisation**:

$$\frac{M\colon i + I}{M^i : I} \quad\longmapsto\quad \frac{M : pi \oplus \mathcal{D}}{M^i : \frac{1}{1-p}\mathcal{D}}$$



The type directly gives the limit of the geometric series:

$$\mathcal{D} + p\mathcal{D} + p^2\mathcal{D} + p^3\mathcal{D} + \cdots \;=\; \frac{1}{1-p}\mathcal{D}$$

The fragment

$$M\,;N \;::=\; i \;\mid\; M\,;i\to N \;\mid\; M^i \;\mid\; \mathrm{rnd}\langle x\rangle.\,x$$

is equivalent to **finite Markov chains**.
Typing computes the **return distribution**.

For the general calculus (FMC with probabilistic types)

$$M\,;N \;::=\; x \;\mid\; \mathrm{rnd}\langle x\rangle.\,M \;\mid\; \langle x\rangle.\,M \;\mid\; [N].\,M \;\mid\; i \;\mid\; M\,;i\to N \;\mid\; M^i$$

typing guarantees **almost-sure termination**.

Thank you