# The Functional Machine Calculus

Chris Barrett, Guy McCusker, Willem Heijltjes
University of Bath

# Overview: The Functional Machine Calculus

Part I: **Confluence for reader/writer effects**

- Global state, probabilistic/non-deterministic choice, I/O
- Express both CBN and CBV semantics

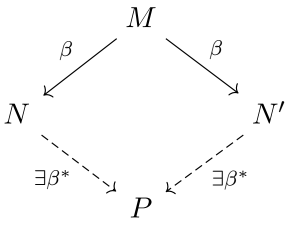
Part II: **Preserves good properties of $\lambda$-calculus:**

- Simple types guarantee strong normalisation
- Cartesian closed categorical semantics (free)
- Domain theoretic semantics

# Problem: Effectful $\lambda$-calculi are Non-confluent

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

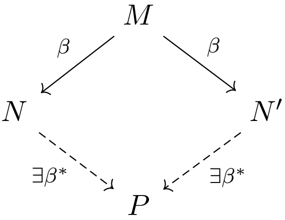
$$(\lambda x.M)N \rightarrow_\beta M\{N/x\}$$



# Problem: Effectful $\lambda$-calculi are Non-confluent

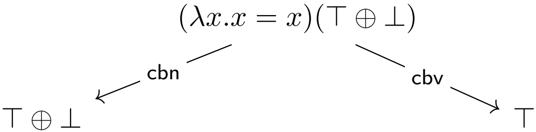
$$M, N \quad ::= \quad x \mid MN \mid \lambda x.M$$

$$(\lambda x.M)N \to_\beta M\{N/x\}$$




$$M, N \quad ::= \quad \ldots \mid M \oplus N$$

$$M \oplus N \to \begin{cases} M & 50\% \\ N & 50\% \end{cases}$$



# Part I

Desiderata: a **confluent calculus** which can express both **CBN and CBV semantics of reader/writer effects**

Solution: generalize the $\lambda$-calculus with

- **Sequencing**: CBN and CBV translations which preserve operational semantics
- **Locations**: Effects and higher-order computation unified: operationally, syntactically, equationally (beta)

# $\lambda$-calculus: Operational Semantics

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

$$S \ ::= \ \epsilon \ | \ S{\cdot}M$$

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

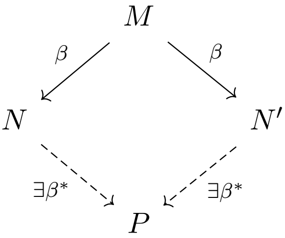
# $\lambda$-calculus: $\beta$-reduction

$$
\begin{array}{lcccccc}
M,\,N & ::= & x & \mid & M\,N & \mid & \lambda x.M \\
M,\,N & ::= & x.\star & \mid & [N].\,M & \mid & \langle x\rangle.\,M
\end{array}
$$

$$S \;::=\; \epsilon \;\mid\; S{\cdot}M$$

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

$$[N].\,\langle x\rangle.\,M \;\;\rightarrow_\beta\;\; \{N/x\}M$$

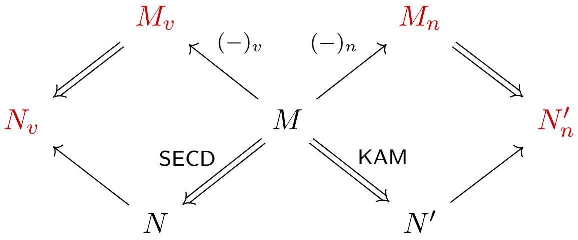


# Sequential $\lambda$-calculus: Operational Semantics

$$
\begin{array}{lcccccccc}
M, N & ::= & & & x & \mid & M\,N & \mid & \lambda x.M \\
M, N & ::= & & & x.\star & \mid & [N].\,M & \mid & \langle x\rangle.\,M \\
M, N & ::= & \star & \mid & x.\,M & \mid & [N].\,M & \mid & \langle x\rangle.\,M
\end{array}
$$

$$
S \;::=\; \epsilon \;\mid\; S\cdot M
$$

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



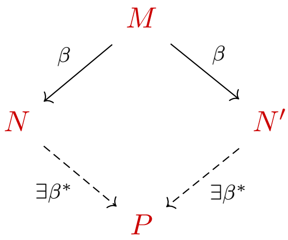
# Sequential $\lambda$-calculus: $\beta$-reduction

$$
\begin{array}{lclcccccc}
M,\,N & ::= & & & x & \mid & M\,N & \mid & \lambda x.M \\
M,\,N & ::= & & & x.\star & \mid & [N].\,M & \mid & \langle x\rangle.\,M \\
M,\,N & ::= & \star & \mid & x.\,M & \mid & [N].\,M & \mid & \langle x\rangle.\,M
\end{array}
$$

$$
S \;::=\; \epsilon \;\mid\; S{\cdot}M
$$

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

$$
[N].\,\langle x\rangle.\,M \;\;\rightarrow_\beta\;\; \{N/x\}M
$$



# Sequencing and Substitution

Capture-avoiding **composition** or **sequencing** $N\,;M$:

$$\begin{aligned}
\star\;;M &= M \\
[P].\,N\,;M &= [P].\,(N\,;M) \\
x.\,N\,;M &= x.\,(N\,;M) \\
\langle x\rangle.\,N\,;M &= \langle x\rangle.\,(N\,;M) && x \notin \mathsf{fv}(M)
\end{aligned}$$

Capture-avoiding **substitution** $\{N/x\}M$:

$$\begin{aligned}
\{P/x\}\star &= \star \\
\{P/x\}x.\,M &= P\,;\{P/x\}M \\
\{P/x\}y.\,M &= y.\,\{P/x\}M && x \neq y \\
&\dots
\end{aligned}$$

# Sequential $\lambda$-terms as Stack Transformers

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

if $\dfrac{(\ R\ ,\ M\ )}{(\ S\ ,\ \star\ )}$ and $\dfrac{(\ S\ ,\ N\ )}{(\ T\ ,\ \star\ )}$ then $\dfrac{\dfrac{(\ R\ ,\ M\ ;N\ )}{(\ S\ ,\ N\ )}}{(\ T\ ,\ \star\ )}$

# Example: Sequential $\lambda$-terms

$$\langle x\rangle.\,\langle y\rangle.\,[y].\,[x].\,\star \qquad \langle x\rangle.\,\langle y\rangle.\,[x].\,[y].\,\star$$

$$\langle x\rangle.\,[x].\,[x] \qquad \langle x\rangle$$

$$\langle f\rangle.\,f.\,f \qquad \langle x\rangle.\,[[x]]$$

# Translating the Call-by-Value $\lambda$-calculus

$$x_v \triangleq [x] \qquad (\lambda x.M)_v \triangleq [\langle x \rangle .\, M_v] \qquad (MN)_v \triangleq N_v \,;\, M_v \,;\, \langle x \rangle .\, x$$

E.g. $(\lambda x.M)N$ translates and runs as:

$$\begin{array}{l}
(\ \epsilon \ ,\ N_v \,;\, [\langle x \rangle .\, M_v] \,;\, \langle y \rangle .\, y\ ) \\
\hline\hline
(\ N' \ ,\ [\langle x \rangle .\, M_v] \,;\, \langle y \rangle .\, y\ ) \\
\hline
(\ N' \cdot \langle x \rangle .\, M_v \ ,\ \langle y \rangle .\, y\ ) \\
\hline
(\ N' \ ,\ \langle x \rangle .\, M_v\ ) \\
\hline
(\ \epsilon \ ,\ \{N'/x\}M_v\ )
\end{array}$$

where $N_v$ returns value $N'$.

# Sequential $\lambda$-calculus: Operational Semantics

$$
\begin{array}{lcccccccc}
M,\,N & ::= & & & x & \mid & M\,N & \mid & \lambda x.M \\
M,\,N & ::= & & & x.\star & \mid & [N].\,M & \mid & \langle x\rangle.\,M \\
M,\,N & ::= & \star & \mid & x.\,M & \mid & [N].\,M & \mid & \langle x\rangle.\,M
\end{array}
$$

$$
S \;::=\; \epsilon \;\mid\; S{\cdot}M
$$

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

# Sequential $\lambda$-calculus: Operational Semantics

$$
\begin{array}{lcccccccc}
M, N & ::= & & & x & \mid & M\,N & \mid & \lambda x.M \\
M, N & ::= & & & x.\star & \mid & [N].\,M & \mid & \langle x\rangle.\,M \\
M, N & ::= & \star & \mid & x.\,M & \mid & [N].\,M & \mid & \langle x\rangle.\,M
\end{array}
$$

$$
S \;::=\; \epsilon \;\mid\; S\cdot M
$$

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

| | Operational | Equational |
|---|---|---|
| @/$\lambda$ | push/pop | $\beta$ |
| input | | |
| output | | |
| state | | |

# Functional Machine Calculus: Operational Semantics

$$
\begin{array}{lllllllll}
M,N & ::= & & & x & \mid & MN & \mid & \lambda x.M \\
M,N & ::= & & & x.\star & \mid & [N].M & \mid & \langle x\rangle.M \\
M,N & ::= & \star & \mid & x.M & \mid & [N].M & \mid & \langle x\rangle.M \\
M,N & ::= & \star & \mid & x.M & \mid & [N]a.M & \mid & a\langle x\rangle.M
\end{array}
$$

$$
S ::= \epsilon \mid S\cdot M \qquad S_A ::= \{ S_a \mid a \in A \}
$$

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

| | Operational | Equational |
|---|---|---|
| @/$\lambda$ | push/pop | $\beta$ |
| input | pop from read-only stream | $\beta\pi$ |
| output | push to write-only stream | $\beta\pi$ |
| state | push/pop on stack of depth one | $\beta\pi$ |

$\beta\pi$-reduction captures algebraic effect equations

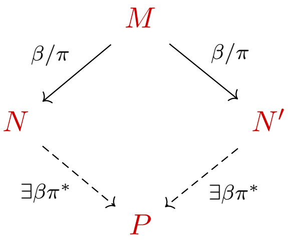
# Functional Machine Calculus: $\beta\pi$-reduction

$$
\begin{array}{llccccccc}
M,\,N & ::= & & & x & \mid & M\,N & \mid & \lambda x.M \\
M,\,N & ::= & & & x.\star & \mid & [N].\,M & \mid & \langle x\rangle.\,M \\
M,\,N & ::= & \star & \mid & x.\,M & \mid & [N].\,M & \mid & \langle x\rangle.\,M \\
M,\,N & ::= & \star & \mid & x.\,M & \mid & [N]a.\,M & \mid & a\langle x\rangle.\,M
\end{array}
$$

$$
S \;::=\; \epsilon \;\mid\; S\cdot M \qquad S_A \;::=\; \{\, S_a \mid a \in A\}
$$

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

$$
\begin{array}{lll}
[N]a.\,a\langle x\rangle.\,M & \rightarrow_\beta & \{N/x\}M \\
[N]a.\,b\langle x\rangle.\,M & \rightarrow_\pi & b\langle x\rangle.\,[N]a.\,M
\end{array}
$$



# Example: Encoding Effects

| rand | set $c$ | get $c$ | write |
|---|---|---|---|
| $\mathsf{rnd}\langle x\rangle.\,[x]$ | $\langle x\rangle.\,c\langle\_\rangle.\,[x]c$ | $c\langle x\rangle.\,[x]c.\,[x]$ | $\langle x\rangle.\,[x]\mathsf{out}$ |
| $\mathsf{rnd}(\mathbb{Z})\Rightarrow\mathbb{Z}$ | $\mathbb{Z}c(\mathbb{Z})\Rightarrow c(\mathbb{Z})$ | $c(\mathbb{Z})\Rightarrow c(\mathbb{Z})\mathbb{Z}$ | $\mathbb{Z}\Rightarrow\mathsf{out}(\mathbb{Z})$ |

$$\frac{\frac{\frac{(\ S_A\ ;\ S_\lambda\cdot M\ ;\ \epsilon_c\cdot N\ ;\ \langle x\rangle.\,c\langle\_\rangle.\,[x]c.\ )}{(\ S_A\ ;\ S_\lambda\ ;\ \epsilon_c\cdot N\ ;\ c\langle\_\rangle.\,[M]c\ )}}{(\ S_A\ ;\ S_\lambda\ ;\ \epsilon_c\ ;\ [M]c\ )}}{(\ S_A\ ;\ S_\lambda\ ;\ \epsilon_c\cdot M\ ;\ \star\ )}$$

$$\frac{\frac{\frac{(\ S_A\ ;\ S_\lambda\ ;\ \epsilon_c\cdot N\ ;\ c\langle x\rangle.\,[x]c.\,[x]\ )}{(\ S_A\ ;\ S_\lambda\ ;\ \epsilon_c\ ;\ [N]c.\,[N]\ )}}{(\ S_A\ ;\ S_\lambda\ ;\ \epsilon_c\cdot N\ ;\ [N]\ )}}{(\ S_A\ ;\ S_\lambda\cdot N\ ;\ \epsilon_c\cdot N\ ;\ \star\ )}$$

# Example: Encoding CBN and CBV State

$$
\begin{aligned}
a := 3; a := 5 &= [3].\,\mathsf{set}\ a.\,[5].\,\mathsf{set}\ a \\
&\to_\beta [3].\,\langle x\rangle.\,a\langle\_\rangle.\,[x]a.\,[5].\,\langle y\rangle.\,a\langle\_\rangle.\,[y]a \\
&\to_\beta a\langle\_\rangle.\,[3]a.\,[5].\,\langle y\rangle.\,a\langle\_\rangle.\,[y]a \\
&\to_\beta a\langle\_\rangle.\,[3]a.\,a\langle\_\rangle.\,[5]a \\
&\to_\beta a\langle\_\rangle.\,[5]a \\
&= a := 5
\end{aligned}
$$

$$
\begin{aligned}
(a := 3; (\lambda x.!a)(a := 5; M))_n &= [3].\,\mathsf{set}\ a.\,[[5].\,\mathsf{set}\ a.\,M_n].\,\langle x\rangle.\,\mathsf{get}\ a \\
&\to_\beta [3].\,\mathsf{set}\ a.\,\mathsf{get}\ a \\
&= a := 3; !a
\end{aligned}
$$

$$
\begin{aligned}
(a := 3; (\lambda x.!a)(a := 5; M))_v &= [3].\,\mathsf{set}\ a.\,[5].\,\mathsf{set}\ a.\,M_v\,;[\langle x\rangle.\,\mathsf{get}\ a].\,\langle f\rangle.\,f \\
&\to_\beta [3].\,\mathsf{set}\ a.\,[5].\,\mathsf{set}\ a.\,M_v\,;\langle x\rangle.\,\mathsf{get}\ a \\
&\to_\beta^* [3].\,\mathsf{set}\ a.\,[5].\,\mathsf{set}\ a.\,\mathsf{get}\ a \\
&= a := 5; !a
\end{aligned}
$$

# Part II: Overview

Part I: **Confluence for reader/writer effects**

- Sequencing: express both CBN and CBV behaviour
- Locations: effects and higher-order computation unified: operationally, syntactically, equationally (beta)

Part II: **Preserving good properties of $\lambda$-calculus**

- Simple types guarantee strong normalisation
- Categorical semantics
- Domain theoretic semantics

# Simply Typed Functional Machine Calculus

$$\tau ::= \overleftarrow{\sigma}_A \Rightarrow \overrightarrow{\tau}_A \mid \alpha \in \Sigma \qquad \overrightarrow{\tau} ::= \tau_n \dots \tau_1 \qquad \overrightarrow{\tau}_A ::= \{\overrightarrow{\tau}_a \mid a \in A\}$$

$$a_1(\sigma_1) \dots a_n(\sigma_n) \Rightarrow b_1(\tau_1) \dots b_m(\tau_m) \qquad\qquad a(\sigma)\, b(\tau) \sim b(\tau)\, a(\sigma)$$

$$\frac{}{\Gamma \vdash \star : \overleftarrow{\tau}_A \Rightarrow \overrightarrow{\tau}_A}\ \star \qquad \frac{\Gamma \vdash N : \rho \qquad \Gamma \vdash M : a(\rho)\, \overleftarrow{\sigma}_A \Rightarrow \overrightarrow{\tau}_A}{\Gamma \vdash [N]a.\, M : \overleftarrow{\sigma}_A \Rightarrow \overrightarrow{\tau}_A}\ \textsf{app}$$

$$\frac{\Gamma,\, x : \rho \vdash M : \overleftarrow{\sigma}_A \Rightarrow \overrightarrow{\tau}_A}{\Gamma \vdash a\langle x \rangle.\, M : a(\rho)\, \overleftarrow{\sigma}_A \Rightarrow \overrightarrow{\tau}_A}\ \textsf{abs} \qquad \frac{\Gamma,\, x : \overleftarrow{\rho}_A \Rightarrow \overrightarrow{\sigma}_A \vdash \quad M : \overleftarrow{\sigma}_A\, \overleftarrow{\tau}_A \Rightarrow \overrightarrow{\upsilon}_A}{\Gamma,\, x : \overleftarrow{\rho}_A \Rightarrow \overrightarrow{\sigma}_A \vdash x.\, M : \overleftarrow{\rho}_A\, \overleftarrow{\tau}_A \Rightarrow \overrightarrow{\upsilon}_A}\ \textsf{var}$$

# Termination and Strong Normalisation

**Theorem : Successful Termination**

$$\vdash M : \overleftarrow{\sigma}_A \Rightarrow \overrightarrow{\tau}_A \quad \text{then} \quad \forall S_A : \overrightarrow{\sigma}_A \,.\; \exists T_A : \overrightarrow{\tau}_A \,.\; \frac{(\,S_A \;,\; M\,)}{(\,T_A \;,\; \star\,)}$$

**Theorem : Strong Normalisation**

$$\Gamma \vdash M \rightarrow_{\beta\pi} M' \quad \text{implies} \quad \lfloor [\![ M ]\!] \rfloor >_{\mathbb{N}} \lfloor [\![ M' ]\!] \rfloor$$

# Categorical Semantics

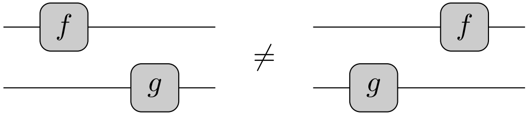
**Definition : FMC**$(\Sigma)/$ ??

- **Objects**: memory types $\vec{\tau}_A$ over $\Sigma_0$,
- **Morphisms**: closed terms $M : \overleftarrow{\sigma}_A \Rightarrow \vec{\tau}_A$ over $\Sigma_1$, mod ??
- **Composition**: $M\,;N$ with identity $\star$

# Categorical Semantics: Pre-monoidal ($\beta\star$)

**Definition : FMC$(\Sigma)/\beta\star$**

- **Objects**: memory types $\vec{\tau}_A$ over $\Sigma_0$,
- **Morphisms**: closed terms $M : \overleftarrow{\sigma}_A \Rightarrow \vec{\tau}_A$ over $\Sigma_1$, mod $\beta\star$
- **Composition**: $M\,;N$ with identity $\star$



# Categorical Semantics: Cartesian Closed ($\sim$)

**Definition : FMC$(\Sigma)/\sim$**

- **Objects**: memory types $\vec{\tau}_A$ over $\Sigma_0$,
- **Morphisms**: closed terms $M : \overleftarrow{\sigma}_A \Rightarrow \vec{\tau}_A$ over $\Sigma_1$, mod $\sim$
- **Composition**: $M\,;N$ with identity $\star$

**Definition : Machine Equivalence ($\sim$)**

Terms equivalent if equivalent inputs result in equivalent outputs:

$$M \sim M' : \overleftarrow{\sigma}_A \Rightarrow \vec{\tau}_A \;\triangleq\; \forall\, S_A {\sim} S'_A : \vec{\sigma}_A.\ (S_A, M)\Downarrow \sim (S'_A, M')\Downarrow : \vec{\tau}_A$$

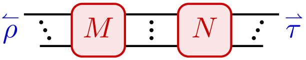
**Theorem : Cartesian Closure**

FMC$(\Sigma)/\sim$ is Cartesian closed, assuming **uniformity** of locations.

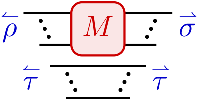
The free functor CCC$(\Sigma)$ to FMC$(\Sigma)/\sim$ is the CBV translation

# Cartesian Equipment: String Diagrams

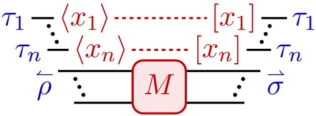

$M : \overleftarrow{\rho} \Rightarrow \overrightarrow{\sigma}$



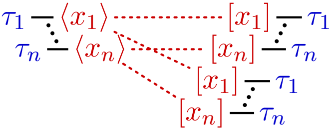

$M \mathbin{;} N : \overleftarrow{\rho} \Rightarrow \overrightarrow{\tau}$



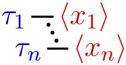

$M : \overleftarrow{\rho}\overleftarrow{\tau} \Rightarrow \overrightarrow{\tau}\overrightarrow{\sigma}$




$\langle\overleftarrow{x}\rangle.\,(M \mathbin{;} [\overrightarrow{x}]) : \overleftarrow{\tau}\overleftarrow{\rho} \Rightarrow \overrightarrow{\sigma}\overrightarrow{\tau}$



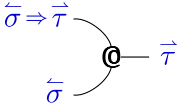

$\langle\overleftarrow{x}\rangle.\,[\overrightarrow{x}].\,[\overrightarrow{x}] : \overleftarrow{\sigma} \Rightarrow \overrightarrow{\sigma}\overrightarrow{\sigma}$



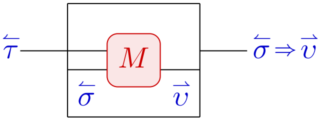

$\langle\overleftarrow{x}\rangle : \overleftarrow{\sigma} \Rightarrow$

# Cartesian Closed Equipment: String Diagrams

$\langle x \rangle . \, x$:



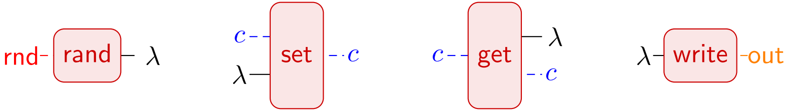

$\langle x \rangle . \, [[x] . \, M]$:



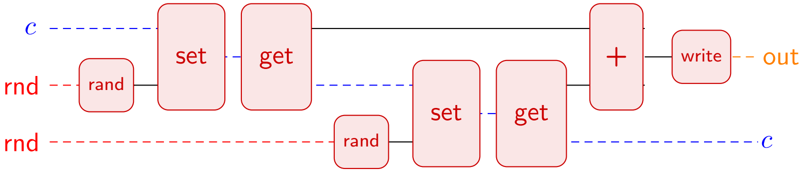

Functorial String Diagrams for Reverse-Mode Automatic Differentiation,
Rewriting for Monoidal Closed Categories,
Alvarez-Picallo, Ghica, Sprunger, Zanasi

# String Diagrams: Effectful Terms

| rand | set $c$ | get $c$ | write |
|---|---|---|---|
| $\mathsf{rnd}\langle x\rangle.\,[x]$ | $\langle x\rangle.\,c\langle \_ \rangle.\,[x]c$ | $c\langle x\rangle.\,[x]c.\,[x]$ | $\langle x\rangle.\,[x]\mathsf{out}$ |
| $\mathsf{rnd}(\mathbb{Z}) \Rightarrow \mathbb{Z}$ | $\mathbb{Z}c(\mathbb{Z}) \Rightarrow c(\mathbb{Z})$ | $c(\mathbb{Z}) \Rightarrow c(\mathbb{Z})\mathbb{Z}$ | $\mathbb{Z} \Rightarrow \mathsf{out}(Z)$ |

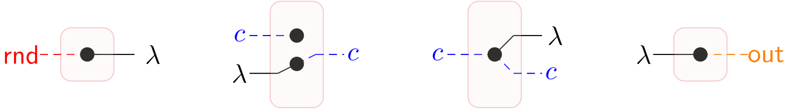



rand ; set ; get ; rand ; set ; get ; + ; write

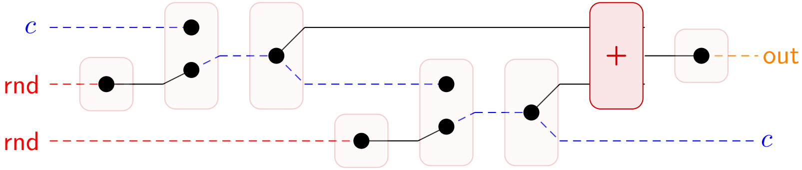


# String Diagrams: Effectful Terms

| rand | set $c$ | get $c$ | write |
|---|---|---|---|
| $\mathsf{rnd}\langle x\rangle.\,[x]$ | $\langle x\rangle.\,c\langle\_\rangle.\,[x]c$ | $c\langle x\rangle.\,[x]c.\,[x]$ | $\langle x\rangle.\,[x]\mathsf{out}$ |
| $\mathsf{rnd}(\mathbb{Z})\Rightarrow\mathbb{Z}$ | $\mathbb{Z}c(\mathbb{Z})\Rightarrow c(\mathbb{Z})$ | $c(\mathbb{Z})\Rightarrow c(\mathbb{Z})\mathbb{Z}$ | $\mathbb{Z}\Rightarrow\mathsf{out}(Z)$ |

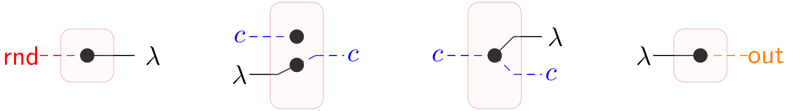



rand ; set ; get ; rand ; set ; get ; + ; write

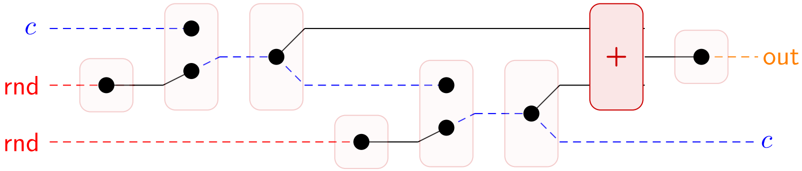


# String Diagrams: Effectful Terms

| rand | set $c$ | get $c$ | write |
|---|---|---|---|
| $\mathsf{rnd}\langle x\rangle.\,[x]$ | $\langle x\rangle.\, c\langle \_ \rangle.\, [x]c$ | $c\langle x\rangle.\, [x]c.\, [x]$ | $\langle x\rangle.\, [x]\mathsf{out}$ |
| $\mathsf{rnd}(\mathbb{Z}) \Rightarrow \mathbb{Z}$ | $\mathbb{Z}c(\mathbb{Z}) \Rightarrow c(\mathbb{Z})$ | $c(\mathbb{Z}) \Rightarrow c(\mathbb{Z})\mathbb{Z}$ | $\mathbb{Z} \Rightarrow \mathsf{out}(Z)$ |

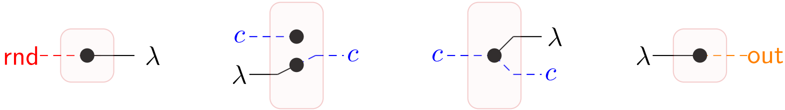



rand ; set ; $c\langle x\rangle.\, [x]c.\, [x]$ ; rand ; $\langle z\rangle.\, \langle \_ \rangle c.\, [z]c$ ; get ; $+$ ; write

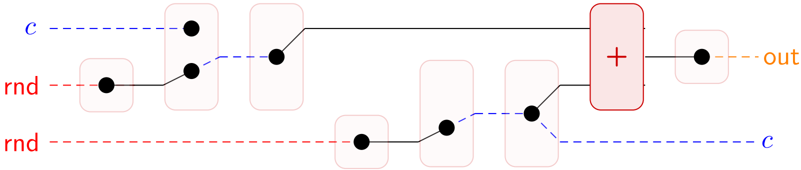


# String Diagrams: Effectful Terms

| rand | set $c$ | get $c$ | write |
|---|---|---|---|
| $\mathsf{rnd}\langle x\rangle.\,[x]$ | $\langle x\rangle.\,c\langle\_\rangle.\,[x]c$ | $c\langle x\rangle.\,[x]c.\,[x]$ | $\langle x\rangle.\,[x]\mathsf{out}$ |
| $\mathsf{rnd}(\mathbb{Z})\Rightarrow\mathbb{Z}$ | $\mathbb{Z}c(\mathbb{Z})\Rightarrow c(\mathbb{Z})$ | $c(\mathbb{Z})\Rightarrow c(\mathbb{Z})\mathbb{Z}$ | $\mathbb{Z}\Rightarrow\mathsf{out}(Z)$ |



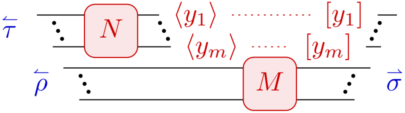

rand ; set ; $c\langle x\rangle.\,[x].\,\mathsf{rand}.\,\langle z\rangle.\,[z]c$ ; get ; + ; write



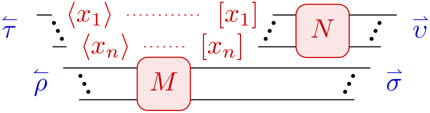
# Categorical Semantics: Free Cartesian Closed ($=_{\text{eqn}}$)

## Definition Equational Theory ($=_{\text{eqn}}$)

$$
\begin{array}{rcl}
\langle x\rangle .\,[x] =_{\star} \star & \qquad & \tau \Rightarrow \tau \\
S\,;\langle \overleftarrow{y}\rangle =_{!} \star & & \Rightarrow \\
S\,;\langle \overleftarrow{y}\rangle .\,[\overrightarrow{y}] .\,[\overrightarrow{y}] =_{\Delta} S\,; S & & \Rightarrow \overrightarrow{\sigma}\,\overrightarrow{\sigma} \\
S\,;\langle \overleftarrow{y}\rangle .\,M .\,[\overrightarrow{y}] =_{\iota} M\,; S & & \overleftarrow{\tau} \Rightarrow \overrightarrow{\upsilon}\,\overrightarrow{\sigma} \\
[M]\,;\langle x\rangle .\,x =_{\beta} M & & \overleftarrow{\tau} \Rightarrow \overrightarrow{\upsilon} \\
S\,;\langle \overleftarrow{x}\rangle .\,[[\overrightarrow{x}] .\,M] =_{\eta} [S\,; M] & & \Rightarrow (\overleftarrow{\tau} \Rightarrow \overrightarrow{\upsilon})
\end{array}
$$




$N .\,\langle \overleftarrow{y}\rangle .\,M .\,[\overrightarrow{y}]$




$\langle \overleftarrow{x}\rangle .\,M .\,[\overrightarrow{x}] .\,N$

# Categorical Semantics: Free Cartesian Closed ($=_{\mathsf{eqn}}$)

**Definition Equational Theory** ($=_{\mathsf{eqn}}$)

$$
\begin{array}{rcll}
\langle x\rangle.\,[x] & =_{\star} & \star & \tau \Rightarrow \tau \\
S\,;\langle \overleftarrow{y}\rangle & =_{!} & \star & \Rightarrow \\
S\,;\langle \overleftarrow{y}\rangle.\,[\overrightarrow{y}].\,[\overrightarrow{y}] & =_{\Delta} & S\,;S & \Rightarrow \overrightarrow{\sigma}\,\overrightarrow{\sigma} \\
S\,;\langle \overleftarrow{y}\rangle.\,M.\,[\overrightarrow{y}] & =_{\iota} & M\,;S & \overleftarrow{\tau} \Rightarrow \overrightarrow{\upsilon}\,\overrightarrow{\sigma} \\
[M]\,;\langle x\rangle.\,x & =_{\beta} & M & \overleftarrow{\tau} \Rightarrow \overrightarrow{\upsilon} \\
S\,;\langle \overleftarrow{x}\rangle.\,[[\overrightarrow{x}].\,M] & =_{\eta} & [S\,;M] & \Rightarrow (\overleftarrow{\tau} \Rightarrow \overrightarrow{\upsilon})
\end{array}
$$

**Theorem : Free Cartesian Closure**

$\mathsf{FMC}(\Sigma)/=_{\mathsf{eqn}} \;\cong\; \mathsf{CCC}(\Sigma)$, assuming **uniformity** of locations

# Domain Theory: Extending Scott-style Semantics

Interpret $[\![-]\!]$ terms in domain $D$ of **stack transformers**, satisfying domain equation:

$$D \cong D^{\mathbb{N}} \to TD^{\mathbb{N}},$$

**Reflexive**: $D \cong D \to D$
**Sequencing**: $D \times D \to D$ as Kleisli composition
**Locations**: let $T$ be state monad

**Theorem : Soundness**

$$(S, M) \Downarrow (T, N) \quad \text{implies} \quad [\![M]\!]([\![S]\!]) = [\![N]\!]([\![T]\!])$$

**Theorem : Adequecy**

$$\text{If} \quad [\![M]\!]([\![S]\!]) \neq \bot \quad \text{then} \quad \exists T \,.\, (S, M) \Downarrow (T, \star)$$

# Summary and Future Work: Functional Machine Calculus

## Summary

- An operational refinement of $\lambda$-calculus; semantically sensible
- Can express both **CBN and CBV semantics of reader/writer effects**, with **confluent** reduction
- Preserves **good properties**: simple types, strong normalisation, categorical and domain theoretic semantics

## Future Work

- Weaker type systems for effects, and non-uniform locations
- Extensions: local state, process calculus, stream processes, sums, co/recursion, ...
- **String diagrams**: Frobenius, interacting Hopf, ...