The Functional Machine Calculus

Semantics

Chris Barrett, Guy McCusker, Willem Heijltjes
University of Bath

Overview: Functional Machine Calculus

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

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

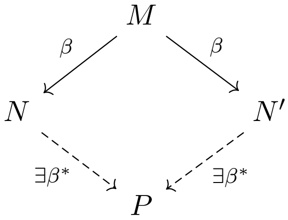
Part II: **Preserves good properties of λ-calculus:**

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

Problem: Effectful λ-calculi are Non-confluent

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

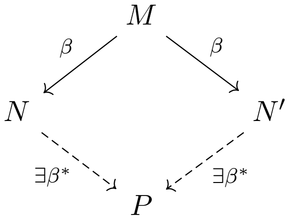
$$(\lambda x.M)N \to_\beta M\{N/x\}$$



Problem: Effectful λ-calculi are Non-confluent

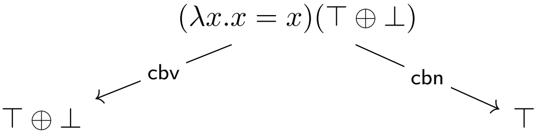
$$M, N \quad ::= \quad x \;\mid\; MN \;\mid\; \lambda x.M$$

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



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

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



Part I: Overview

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

Solution: generalize the λ-calculus with

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

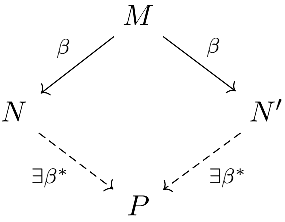
λ-calculus

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

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

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

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



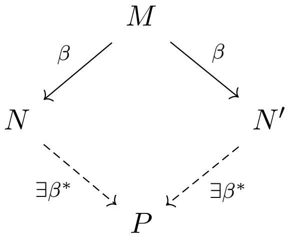
λ-calculus

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

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

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

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



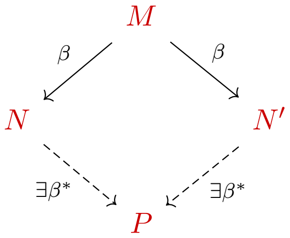
Sequential λ-calculus

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

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

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

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

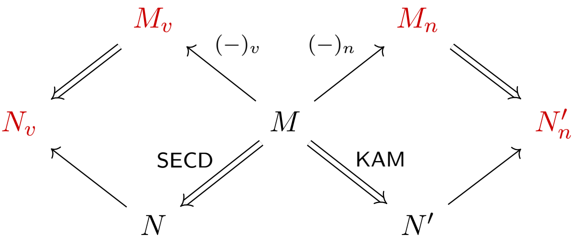


Sequential λ-calculus

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

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

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



Sequencing and Substitution

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

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

Sequential λ-calculus

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

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

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

Functional Machine Calculus

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

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

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

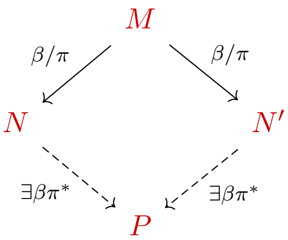
Functional Machine Calculus

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

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

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

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



Functional Machine Calculus

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

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

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

	Operational	Equational
@/λ	push/pop	β
input		
output		
state		

Functional Machine Calculus

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

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

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

	Operational	Equational
@/λ	push/pop	β
input	pop from stream	$\beta\pi$
output	push to stream	$\beta\pi$
state	push/pop on stack of depth one	$\beta\pi$

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

Example: Encoding Effects

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

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

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

Part II: Overview

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

- Sequencing: express both CBN and CBV behaviour
- Locations: unify operational semantics, syntax, reduction of effects and higher-order computation – recovering confluence

Part II: **Preserving good properties of λ-calculus**

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

Simple Types and Termination

$$\tau \ ::= \ \overleftarrow{\sigma}_A \Rightarrow \overrightarrow{\tau}_A \mid \alpha \in \Sigma_0 \qquad \overrightarrow{\tau} \ ::= \ \tau_n \ldots \tau_1 \qquad \overrightarrow{\tau}_A \ ::= \ \{\overrightarrow{\tau}_a \mid a \in A\}$$

Theorem : Successful Termination

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

Strong Normalisation: Gandy-style Proof

Definition : Interpretation $[\![-]\!]$ of types and terms

$$[\![\overleftarrow{\sigma}_A \Rightarrow \overrightarrow{\tau}_A]\!] = \{\textbf{non-strict}\text{ monotone } f : [\![\overrightarrow{\sigma}_A]\!] \to \mathbb{N} \times [\![\overrightarrow{\tau}_A]\!]\}$$

with extensional partial order

Extract measure: $\lfloor - \rfloor : [\![\tau]\!] \to \mathbb{N}$

Theorem : Strong Normalisation

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

Categorical Semantics

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

- **Objects**: memory types $\vec{\tau}_A$ over Σ_0,
- **Morphisms**: closed terms $M : \overleftarrow{\sigma}_A \Rightarrow \vec{\tau}_A$ over Σ_1, mod ??
- **Composition**: $M\,;N$ with identity $\star$

Categorical Semantics

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

- **Objects**: memory types $\vec{\tau}_A$ over Σ_0,
- **Morphisms**: closed terms $M : \overleftarrow{\sigma}_A \Rightarrow \vec{\tau}_A$ over Σ_1, mod $\sim$
- **Composition**: $M\,;N$ with identity $\star$

Definition : Machine Equivalence ($\sim$)

Terms equivalent if equivalent inputs result in equivalent outputs:

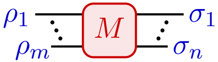
$$M \sim M' : \overleftarrow{\sigma}_A \Rightarrow \vec{\tau}_A \;\triangleq\; \forall\, S_A \sim S'_A : \vec{\sigma}_A.\ (S_A, M)\Downarrow \sim (S'_A, M')\Downarrow : \vec{\tau}_A$$

Theorem : Cartesian Closure

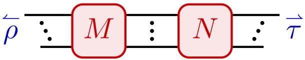
FMC$(\Sigma)/\sim$ is Cartesian closed.

Type system assumes **uniformity** of locations

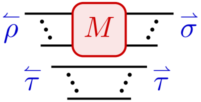
Cartesian Category: String Diagrams



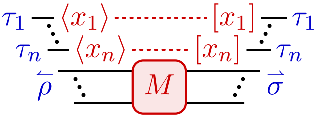
$M : \overleftarrow{\rho} \Rightarrow \overrightarrow{\sigma}$



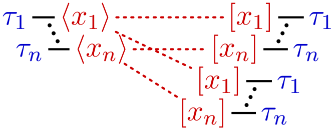
$M \,;\, N : \overleftarrow{\rho} \Rightarrow \overrightarrow{\tau}$



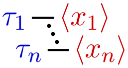
$M : \overleftarrow{\rho}\overleftarrow{\tau} \Rightarrow \overrightarrow{\tau}\overrightarrow{\sigma}$



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



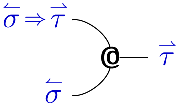
$\langle\overleftarrow{x}\rangle.\,[\overrightarrow{x}].\,[\overrightarrow{x}] : \overleftarrow{\sigma} \Rightarrow \overrightarrow{\sigma}\overrightarrow{\sigma}$



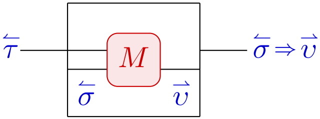
$\langle\overleftarrow{x}\rangle : \overleftarrow{\sigma} \Rightarrow$

Cartesian Closed Category: String Diagrams

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

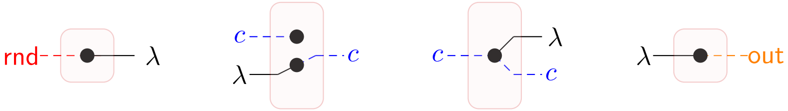


$\langle x \rangle .\, [[x].\, M]$:

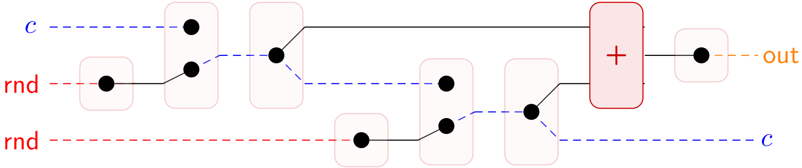


Effectful String Diagrams

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



Categorical Semantics

We can define an equational theory $=_{\text{eqn}} \subset (\sim)$ such that:

Definition : FMC$(\Sigma)/=_{\text{eqn}}$

- **Objects**: memory types $\vec{\tau}_A$ over Σ_0,
- **Morphisms**: closed terms $M : \overleftarrow{\sigma}_A \Rightarrow \vec{\tau}_A$ over Σ_1, mod $=_{\text{eqn}}$
- **Composition**: $M\,;N$ with identity $\star$

Theorem : Free Cartesian Closure

FMC$(\Sigma)/=_{\text{eqn}}$ is the free Cartesian closed category over Σ

Type system assumes **uniformity** of locations

Domain Theory: Extending Scott-style Semantics

Interpret terms in domain D of **stack transformers**:

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

Sequencing: $D \times D \to D$ as Kleisli composition

$$\begin{aligned} D \text{ reflexive} : D &\cong D^{\mathbb{N}} \to T D^{\mathbb{N}} \\ &\cong (D^{\mathbb{N}} \times D) \to T D^{\mathbb{N}} \\ &\cong D \to (D^{\mathbb{N}} \to T D^{\mathbb{N}}) \\ &\cong D \to D \end{aligned}$$

Locations: let T be state monad

Summary: Functional Machine Calculus

- An operational refinement of λ-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