Algebraic Foundations for Effect-Dependent Optimisations

Ohad Kammar
<ohad.kammar@ed.ac.uk>

Gordon Plotkin
<gdp@ed.ac.uk>

Functional Programming Lab Seminar
University of Nottingham
January 11, 2012

To appear in POPL'12

Laboratory for Foundations of Computer Science
Association for Computing Machinery
THE ROYAL SOCIETY
THE UNIVERSITY of EDINBURGH informatics

Gifford-style Types and Effects

Effect systems

$$\textbf{if true then } x := 1$$
$$\textbf{else } x := \textbf{deref}(y)$$

Gifford-style Types and Effects

Effect systems

$$\vdash \textbf{if true then } \mathtt{x} := 1 \quad \textbf{else } \mathtt{x} := \textbf{deref}(\mathtt{y}) : () \,!\, \underbrace{\{\mathtt{lookup}, \mathtt{update}\}}_{\varepsilon}$$

$$\Gamma \vdash M_i : A_i \,!\, \varepsilon_i$$

Effect-dependent optimisations [Benton et al.]

$$\varepsilon_i \subseteq \{\mathtt{lookup}\} \implies \begin{array}{c} \mathtt{let}\ x = M_1\ \mathtt{in}\ (\mathtt{let}\ y = M_2\ \mathtt{in}\ N) \\ = \\ \mathtt{let}\ y = M_2\ \mathtt{in}\ (\mathtt{let}\ x = M_1\ \mathtt{in}\ N) \end{array}$$

Problem

Difficulty

Change language or effects $\implies$ restart from scratch (useful craft).

- Duplicated effort
- Mix routine and important issues

Solution

General semantic account of effect type systems (science).

Prospect

Tools, methods and automatic support (engineering).

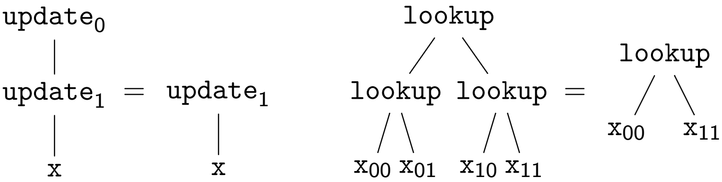
Plan

Tool: algebraic theory of effects

An interface to effects:

Effect operations Σ e.g.: `lookup` : 2, `update` : 1&2

Effect equations E e.g.:



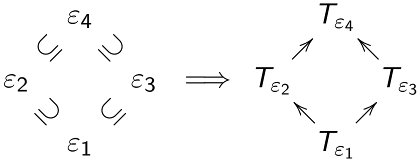
Marriage of effects and monads

Observation [Wadler]

Change notation:

$$\Gamma \vdash M : A \,!\, \varepsilon \implies \Gamma \vdash M : T_{\varepsilon} A$$

T_{ε} behaves like a monad.



Annotation effects as effect operations

Key Observation

$\varepsilon = \{\texttt{lookup}, \texttt{update}\}$ as an algebraic **signature**.

Change Perspective

View T_ε as $\langle \Sigma_\varepsilon, E_\varepsilon \rangle$

Choose $\Sigma_\varepsilon = \varepsilon$

Wonder $E_\varepsilon = ?$

Bird's Eye

monadic source language

$\Downarrow$

multi-monadic intermediate language

$\Downarrow$

semantics

$\Downarrow$

optimisations (logic)

Talk Structure

- Source language
- Intermediate representation (IR) language
- Semantics
  - Validating optimisations
  - Constructing IR models
- Optimisations
- Conclusions

Bird's Eye

monadic source language

$\Downarrow$

multi-monadic intermediate language

$\Downarrow$

semantics

$\Downarrow$

optimisations (logic)

Source syntax

Signature

$\Sigma = \{op : a\&p\}$ parametrises the language.

Example

State: $\texttt{lookup} : 2\ (\texttt{lookup} : 2\&1)$, $\quad \texttt{update} : 1\&2$

Exceptions: $\texttt{DivideByZero} : 0$

Input: $\texttt{input} : 128$, $\texttt{output} : 1\&128$

Already $\mathbf{2^5} = \mathbf{32}$ different languages!

Source syntax

Types and terms

$$
\begin{array}{rcl}
A, B, \ldots & ::= & \mathbf{n} \mid A \to B \mid TA \\
M, N, \ldots & ::= & x \mid i \mid \lambda x.M \mid M\,N \\
& & \mid \mathtt{return}\, M \mid x \leftarrow M; N \\
& & \mid \mathtt{op}_M N
\end{array}
$$

Source syntax

Type system

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M} \qquad \frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\,N : B}$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathtt{return}\, M : TA} \qquad \frac{\Gamma \vdash M : TA \qquad \Gamma, x : A \vdash N : TB}{\Gamma \vdash x \leftarrow M; N : TB}$$

(cont.)

Source syntax

Type system

(contd.)

$$\frac{\Gamma \vdash M : \mathbf{p} \qquad \Gamma \vdash N : \mathbf{a} \to TB}{\Gamma \vdash \mathtt{op}_M N : TB}\ \mathtt{op} : a\&p$$

Example

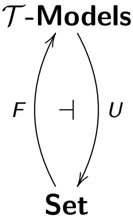
$$\vdash \mathtt{lookup}(\lambda x.\mathtt{update}_0(\lambda\_.\,\mathtt{return}\,x)) : T\mathbf{2}$$

Source semantics

Eilenberg-Moore adjunction

A model is a theory $\mathcal{T} = \langle \Sigma, E \rangle$

Derive an adjunction $F \dashv U$:



Derive a strong monad $Tx := UFx$

Bird's Eye

monadic source language

$\Downarrow$

multi-monadic intermediate language

$\Downarrow$

semantics

$\Downarrow$

optimisations (logic)

IR syntax

Types and terms

$$
\begin{array}{rcll}
A, B, \ldots & ::= & \mathbf{n} \mid A \to B & \mid T_\varepsilon A \\
M, N, \ldots & ::= & x \mid i \mid \lambda x.M & \mid M\, N \\
 & & \mid \texttt{return}_\varepsilon\, M & \mid x \leftarrow M; N \\
 & & \mid \texttt{op}_M N & \mid \texttt{coerce}_{\varepsilon \subseteq \varepsilon'} M
\end{array}
$$

where $\varepsilon, \varepsilon' \subseteq \Sigma$

Type system

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M} \qquad \frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\,N : B}$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathtt{return}_\varepsilon\, M : T_\varepsilon A} \qquad \frac{\Gamma \vdash M : T_\varepsilon A \qquad \Gamma, x : A \vdash N : T_\varepsilon B}{\Gamma \vdash x \leftarrow M; N : T_\varepsilon B}$$

(cont.)

Type system

(contd.)

$$\frac{\Gamma \vdash M : \mathbf{p} \qquad \Gamma \vdash N : \mathbf{a} \to T_\varepsilon B}{\Gamma \vdash \mathtt{op}_M N : T_\varepsilon B}\ \mathtt{op} : a\&p, \mathtt{op} \in \varepsilon$$

$$\frac{\Gamma \vdash M : T_\varepsilon A}{\Gamma \vdash \mathtt{coerce}_{\varepsilon \subseteq \varepsilon'} M : T_{\varepsilon'} A}$$

Example: higher-order coercion

$$\vdash \lambda f.\lambda x.\mathtt{coerce}_{\varepsilon_2 \subseteq \varepsilon_2'}(f(\mathtt{coerce}_{\varepsilon_1 \subseteq \varepsilon_1'} x)) \\ : (T_{\varepsilon_1'} A \to T_{\varepsilon_2} B) \to (T_{\varepsilon_1} A \to T_{\varepsilon_2'} B)$$

Bird's Eye

monadic source language

$\Downarrow$

multi-monadic intermediate language

$\Downarrow$

semantics

$\Downarrow$

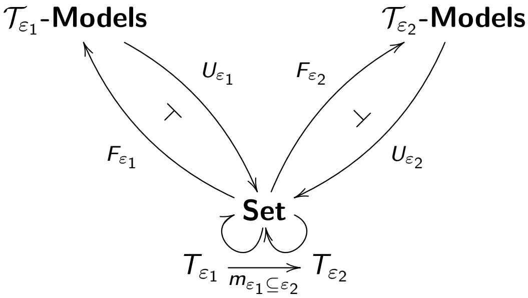
optimisations (logic)

Semantics

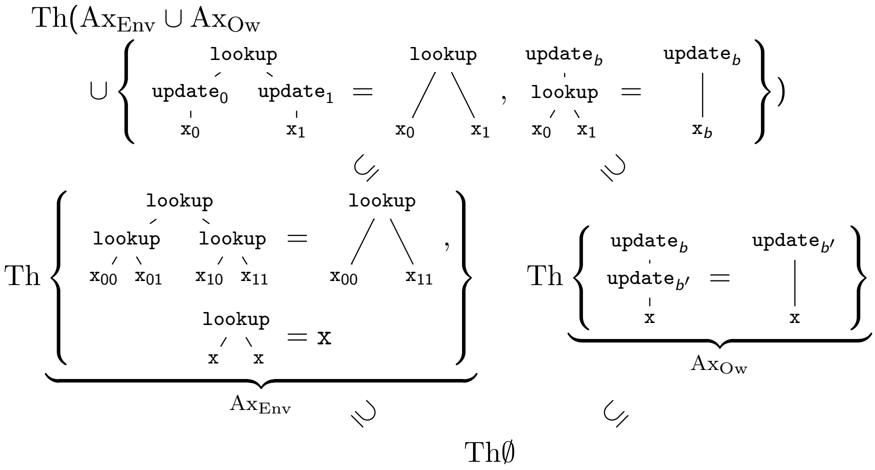
Models

A functorial family of theories: $\mathcal{T}_\varepsilon = \langle \varepsilon, E_\varepsilon \rangle$
with $E_{\varepsilon_1} \subseteq E_{\varepsilon_2}$ whenever $\varepsilon_1 \subseteq \varepsilon_2$.

Derived monads



Example


Th(Ax_Env ∪ Ax_Ow
lookup
update_0
update_1
x_0
x_1
update_b
x_b
Th
lookup
x_00
x_01
x_10
x_11
x
Ax_Env
update_b'
Ax_Ow
Th∅

Derived monads

$$T_{\{\texttt{lookup},\texttt{update}\}}A = (2 \times A)^2$$

$$T_{\{\texttt{lookup}\}}A = A^2 \qquad T_{\{\texttt{update}\}}A = (1 + 2) \times A$$

$$T_{\emptyset}A = A$$

Effect-dependent optimisation

Source: $\mathtt{x} \leftarrow M; \mathtt{return}\, 0 \quad : T\mathbf{1}$

Effect-dependent optimisation

Source: $\mathtt{x} \leftarrow M; \mathtt{return}\, 0 \quad = \quad \mathtt{return}\, 0 \quad : T\mathbf{1}$

IR:

$$\begin{aligned} &\mathtt{x} \leftarrow M; \\ &\mathtt{return}_{\{\mathtt{lookup}\}}\, 0 \quad = \quad \mathtt{return}_{\{\mathtt{lookup}\}}\, 0 \quad : T_{\{\mathtt{lookup}\}}\mathbf{1} \end{aligned}$$

crucial step holds $\forall N : T_{\{\mathtt{lookup}\}}A$, not $\forall N : TA$

Effect-dependent optimisation

Source: $\mathtt{x} \leftarrow M; \mathtt{return}\, 0 \quad = \quad \mathtt{return}\, 0 \quad : T\mathbf{1}$

IR:

$$\begin{aligned}\mathtt{x} \leftarrow M;& \\ \mathtt{return}_{\{\mathtt{lookup}\}}\, 0 \quad &= \quad \mathtt{return}_{\{\mathtt{lookup}\}}\, 0 \quad : T_{\{\mathtt{lookup}\}}\mathbf{1} \\ & \quad\quad \mathtt{return}_{\emptyset}\, 0 \quad : T_{\emptyset}\mathbf{1}\end{aligned}$$

Formalising soundness

Erasure

Erase : IR terms $\to$ source terms

$\mathrm{Erase}(M)$: remove ε's and coercions from M

$$\mathtt{coerce}_{\{\mathtt{lookup}\}}(\mathtt{x} \leftarrow M; \mathtt{return}_{\emptyset}\, 0) \quad \xmapsto{\mathrm{Erase}} \quad \mathtt{x} \leftarrow \mathrm{Erase}(M); \mathtt{return}\, 0$$

Validity

$\mathcal{M}$ a model (source or IR):

$$\mathcal{M} \models M = N \overset{def}{\iff} [\![M]\!] = [\![N]\!] \text{ in } \mathcal{M}$$

Formal soundness

Soundness

For a **source** model $\mathcal{T}$ and IRs $\vdash M, N : T_\varepsilon \mathbf{n}$,
suffices to find an IR model $\mathcal{T}^\sharp$ such that:

$$\mathcal{T}^\sharp \models M = N \implies \mathcal{T} \models \mathrm{Erase}(M) = \mathrm{Erase}(N)$$

Source: $\mathit{Erase}(M)$ $\mathit{Erase}(N) \quad : T\mathbf{n}$

IR: $M = M' = M'' = \ldots = M''' = N \quad : T_\varepsilon \mathbf{n}$

Constructing IR Models

Conservative Restriction Model

Given $\mathcal{T} = \langle \Sigma, E \rangle$, define the IR model $\mathcal{T}^{\mathrm{Cns}}$ by:

$$E|_{\varepsilon} := E \cap (\varepsilon\text{-terms} \times \varepsilon\text{-terms})$$

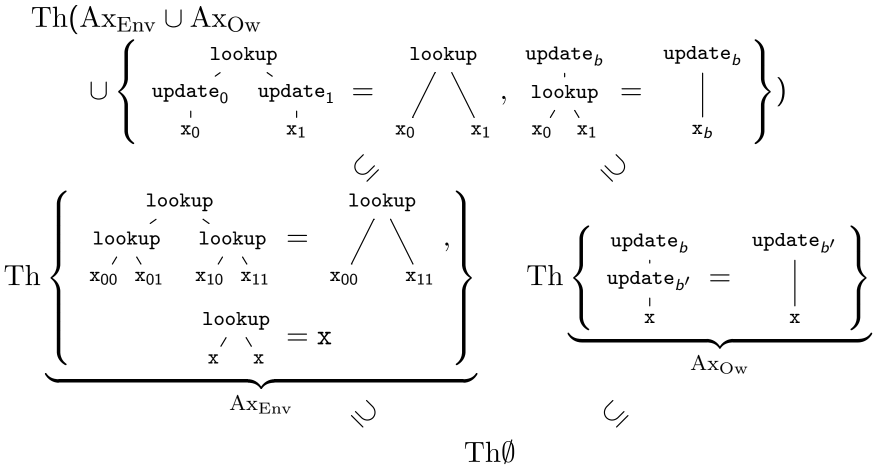
i.e., all derivable E equations between ε-terms.

Theorem

For all $\vdash M, N : T_{\varepsilon}\mathbf{n}$*:*

$$\mathcal{T}^{\mathrm{Cns}} \models M = N \iff \mathcal{T} \models \mathrm{Erase}(M) = \mathrm{Erase}(N)$$

Example Conservative Restriction Model



Global state

Injective monad morphisms

$$T_{\{\texttt{lookup},\texttt{update}\}}A = (2 \times A)^2$$

$$T_{\{\texttt{lookup}\}}A = A^2 \qquad T_{\{\texttt{update}\}}A = (1 + 2) \times A$$

$$T_{\emptyset}A = A$$

Combining theories

Sum

$\mathcal{T}^1 + \mathcal{T}^2 \coloneqq \langle \Sigma_1 + \Sigma_2, \mathrm{Th}(E_1 + E_2) \rangle$

Theorem [Hyland, Plotkin, Power]

Summing with exceptions (resp. input, output) induces the exception (resp. input, output) monad transformer.

Modularity theorem

Idea

Restrictions of $\mathcal{T} = \mathcal{T}^1 \bigcirc \mathcal{T}^2$ in terms of component restrictions.

Theorem

For consistent theories:

$$(\mathcal{T}^1 + \mathcal{T}^2)\big|_{\varepsilon_1 + \varepsilon_2} = \mathcal{T}^1\big|_{\varepsilon_1} + \mathcal{T}^2\big|_{\varepsilon_2}$$

Combining theories

Tensor

$\mathcal{T}^1 + \mathcal{T}^2 := \langle \Sigma_1 + \Sigma_2, \mathrm{Th}((E_1 + E_2) \cup E_{\Sigma_1 \otimes \Sigma_2}) \rangle$ where $E_{\Sigma_1 \otimes \Sigma_2}$ are

$$f(g(x_{00}, x_{01}), g(x_{10}, x_{11})) = g(f(x_{00}, x_{10}), f(x_{01}, x_{11}))$$

Theorem [Hyland, Plotkin, Power]

Tensoring with the global state (resp. environment, overwrite) theory induces the global state (resp. environment, overwrite) monad transformer.

Modularity counter example

Idea

Restrictions of $\mathcal{T} = \mathcal{T}^1 \bigcirc \mathcal{T}^2$ in terms of component restrictions.

Tensor counterexample: Eckmann-Hilton

$$\begin{aligned}(\text{Monoids} \otimes \text{Monoids})_{\{\cdot,1\}+\emptyset} &= \text{Commutative Monoids} \\ &\neq \text{Monoids}|_{\{\cdot,1\}} \otimes \text{Monoids}|_{\emptyset}\end{aligned}$$

$$(1 \cdot_1 x) \cdot_2 (y \cdot_1 1) = (1 \cdot_2 y) \cdot_1 (x \cdot_2 1) \implies x \cdot_2 y = y \cdot_1 x$$
$$(x \cdot_1 1) \cdot_2 (1 \cdot_1 y) = (x \cdot_2 1) \cdot_1 (1 \cdot_2 y) \implies x \cdot_2 y = x \cdot_1 x$$

Pragmatic Modularity Theorems

Tensoring with the global state, environment and overwrite theories is modular. Tensoring with non-determinism is non-modular over ω-CPO.

Axiomatic restriction

Axiomatic Restriction Model

Given $\mathcal{T} = \langle \Sigma, \mathrm{ThAx} \rangle$, define the IR model $\mathcal{T}^{\mathrm{Ax}}$ by:

$$\mathrm{Th}|_{\varepsilon}\,\mathrm{Ax} \coloneqq \mathrm{Th}(\mathrm{Ax} \cap (\varepsilon\text{-terms} \times \varepsilon\text{-terms}))$$

By fiat,

$$\begin{aligned}
\mathrm{Th}|_{\varepsilon_1+\varepsilon_2}\,(\mathrm{Ax}^1 + \mathrm{Ax}^2) &= \mathrm{Th}|_{\varepsilon_1}\,\mathrm{Ax}^1 + \mathrm{Th}|_{\varepsilon_2}\,\mathrm{Ax}^2 \\
\mathrm{Th}|_{\varepsilon_1+\varepsilon_2}\,((\mathrm{Ax}^1 + \mathrm{Ax}^2) \cup E_{\Sigma_1 \otimes \Sigma_2}) &= \mathrm{Th}|_{\varepsilon_1}\,\mathrm{Ax}^1 \otimes \mathrm{Th}|_{\varepsilon_2}\,\mathrm{Ax}^2
\end{aligned}$$

Theorem

For all $\vdash M, N : T_{\varepsilon}\mathbf{n}$:

$$\mathcal{T}^{\mathrm{Ax}} \models M = N \implies \mathcal{T} \models \mathrm{Erase}(M) = \mathrm{Erase}(N)$$

Bird's Eye

monadic source language

$\Downarrow$

multi-monadic intermediate language

$\Downarrow$

semantics

$\Downarrow$

optimisations (logic)

Cataloguing Optimisations

For existing transformations:

- Validate
- Classify
- Generalise

Structural properties

Structural

Bread and butter of optimisation, e.g.

- β, η rules.
- Sequencing.
- Coercion, e.g.:

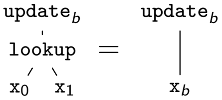
$$\texttt{coerce}_{\varepsilon' \subseteq \varepsilon''}(\texttt{coerce}_{\varepsilon \subseteq \varepsilon'} M) = \texttt{coerce}_{\varepsilon \subseteq \varepsilon''} M$$

Practically: constant propagation, common subexpression elimination, (loop unrolling), etc.

Local algebraic properties

Algebraic

Single equations in $\mathcal{T}_\varepsilon$, e.g.:



become optimisations, e.g.:

$$\texttt{update}_V(\texttt{lookup}(N)) = \texttt{update}_V N\ V$$

i.e., **local** properties of $\mathcal{T}_\varepsilon$.

Global structural properties

Discard: Utilitarian Form

$$\frac{\Gamma \vdash M : T_\varepsilon A \qquad \Gamma \vdash N : T_{\varepsilon'} B}{\mathtt{x} \leftarrow (\mathtt{coerce}_{\varepsilon \subseteq \varepsilon'} M); N = N}$$

Discard: Pristine Form

$$\frac{\Gamma \vdash M : T_\varepsilon A}{\mathtt{x} \leftarrow M; \mathtt{return}_\varepsilon\, 0 = \mathtt{return}_\varepsilon\, 0}$$

(cont.)

Abstract optimisations

(contd.) Discard: $\mathtt{x} \leftarrow M; \mathtt{return}_\varepsilon\, 0 = \mathtt{return}_\varepsilon\, 0$

Categorical Characterisation

$$T_\varepsilon 1 \cong 1$$

Due to Kock, Jacobs, Führmann

Algebraic Characterisation

For all $t(\mathtt{x}_1, \ldots, \mathtt{x}_n)$:

$$\begin{array}{ccc} & t & \\ \swarrow & & \searrow \\ \mathtt{x} & \cdots & \mathtt{x} \end{array} = \mathtt{x}$$

Due to Wraith

This is a **global** property.

Effective dictionary

Knowledge unification

Optimisation	Utilitarian	Pristine	Categorical	Algebraic
⋮	⋮	⋮	⋮	⋮

Effective dictionary

name	utilitarian form	pristine form	abstract side condition	algebraic equivalent	example basic theories
Discard	B F $\dfrac{\Gamma \vdash_{\varepsilon} M : \mathbf{F}_{\varepsilon} A \quad \Gamma \vdash_{\varepsilon'} N : \underline{B}}{(\mathbf{coerce}\, M)\ \mathbf{to}\ x : A.\, N = N}$	B F $\dfrac{\Gamma \vdash_{\varepsilon} M : \mathbf{F}_{\varepsilon} A}{M\ \mathbf{to}\ x : A.\, \mathbf{return}_{\varepsilon} \star = \mathbf{return}_{\varepsilon} \star}$	F $\mathcal{T}_{\varepsilon}$ affine: $\eta^{\varepsilon}_{\mathbb{1}} : \mathbb{1} \to \lvert F_{\varepsilon} \mathbb{1} \rvert$ has a continuous inverse	For all ε-terms t: $t(\mathbf{x}, \ldots, \mathbf{x}) = \mathbf{x}$	read-only state, convex, upper and lower semilattices
Copy	B F $\dfrac{\Gamma \vdash_{\varepsilon} M : \mathbf{F}_{\varepsilon} A \quad \Gamma, x : A, y : A \vdash_{\varepsilon'} N : \underline{B}}{\mathbf{coerce}\, M\ \mathbf{to}\ x : A.\ \mathbf{coerce}\, M\ \mathbf{to}\ y : A.\, N = \mathbf{coerce}\, M\ \mathbf{to}\ x : A.\, N[x/y]}$	B F $\dfrac{\Gamma \vdash_{\varepsilon} M : \mathbf{F}_{\varepsilon} A}{M\ \mathbf{to}\ x : A.\ M\ \mathbf{to}\ y : A.\, \mathbf{return}_{\varepsilon}(x, y) = M\ \mathbf{to}\ x : A.\, \mathbf{return}_{\varepsilon}(x, x)}$	F $\mathcal{T}_{\varepsilon}$ relevant: $\psi_{\varepsilon} \circ \delta = L^{\varepsilon} \delta$	For all ε-terms t: $t(t(\mathbf{x}_{11}, \ldots, \mathbf{x}_{1n}), \ldots, t(\mathbf{x}_{n1}, \ldots, \mathbf{x}_{nn})) = t(\mathbf{x}_{11}, \ldots, \mathbf{x}_{nn})$	exceptions, lifting, read-only state, write-only state
Weak Copy	$\dfrac{\Gamma \vdash_{\varepsilon} M : \mathbf{F}_{\varepsilon} A \quad \Gamma, x : A \vdash_{\varepsilon'} N : \underline{B}}{\mathbf{coerce}\, M\ \mathbf{to}\ x : A.\ \mathbf{coerce}\, M\ \mathbf{to}\ y : A.\, N = \mathbf{coerce}\, M\ \mathbf{to}\ x : A.\, N}$	$\dfrac{\Gamma \vdash_{\varepsilon} M : \mathbf{F}_{\varepsilon} A}{M\ \mathbf{to}\ x : A.\, M = M}$	$\mu^{\varepsilon} \circ \mathrm{str}^{\varepsilon} \circ \delta = \mathrm{id}$	For all ε-terms t: $t(t(\mathbf{x}_1, \ldots, \mathbf{x}_n), \ldots, t(\mathbf{x}_1, \ldots, \mathbf{x}_n)) = t(\mathbf{x}_1, \ldots, \mathbf{x}_n)$	any affine or relevant theory: lifting, exceptions, read-only and write-only state, all three semilattice theories
Swap	B F $\dfrac{\Gamma \vdash_{\varepsilon_1} M_1 : \mathbf{F}_{\varepsilon_1} A_1 \quad \Gamma \vdash_{\varepsilon_2} M_2 : \mathbf{F}_{\varepsilon_2} A_2 \quad \Gamma, x_1 : A_1, x_2 : A_2 \vdash_{\varepsilon'} N}{\mathbf{coerce}\, M_1\ \mathbf{to}\ x_1 : A_1.\ \mathbf{coerce}\, M_2\ \mathbf{to}\ x_2 : A_2.\, N = \mathbf{coerce}\, M_2\ \mathbf{to}\ x_2 : A_2.\ \mathbf{coerce}\, M_1\ \mathbf{to}\ x_1 : A_1.\, N}$	B F $\dfrac{\Gamma \vdash_{\varepsilon_1} M_1 : \mathbf{F}_{\varepsilon_1} A_1 \quad \Gamma \vdash_{\varepsilon_2} M_2 : \mathbf{F}_{\varepsilon_2} A_2}{\mathbf{coerce}\, M_1\ \mathbf{to}\ x_1 : A_1.\ \mathbf{coerce}\, M_2\ \mathbf{to}\ x_2 : A_2.\, \mathbf{return}_{\varepsilon}(x_1, x_2) = \mathbf{coerce}\, M_2\ \mathbf{to}\ x_2 : A_2.\ \mathbf{coerce}\, M_1\ \mathbf{to}\ x_1 : A_1.\, \mathbf{return}_{\varepsilon}(x_1, x_2)}$	F $\mathfrak{T}_{\varepsilon_1 \subseteq \varepsilon}, \mathfrak{T}_{\varepsilon_2 \subseteq \varepsilon}$ commute: $\psi_{\varepsilon} \circ (m^{\varepsilon_1 \subseteq \varepsilon} \times m^{\varepsilon_2 \subseteq \varepsilon}) = \tilde{\psi}_{\varepsilon} \circ (m^{\varepsilon_1 \subseteq \varepsilon} \times m^{\varepsilon_2 \subseteq \varepsilon})$	$\mathfrak{T}_{\varepsilon_1 \subseteq \varepsilon}$ translations commute with $\mathfrak{T}_{\varepsilon_2 \subseteq \varepsilon}$ translations (see tensor equations)	$\mathcal{T}_1 \to \mathcal{T}_1 \otimes \mathcal{T}_2 \leftarrow \mathcal{T}_2$, e.g., distinct global memory cells
Weak Swap	$\dfrac{\Gamma \vdash_{\varepsilon_1} M_1 : \mathbf{F}_{\varepsilon_1} A_1 \quad \Gamma \vdash_{\varepsilon_2} M_2 : \mathbf{F}_{\varepsilon_2} A_2 \quad \Gamma, x_1 : A_1 \vdash_{\varepsilon'} N}{\text{(same as Swap)}}$	$\dfrac{\Gamma \vdash_{\varepsilon_1} M_1 : \mathbf{F}_{\varepsilon_1} A_1 \quad \Gamma \vdash_{\varepsilon_2} M_2 : \mathbf{F}_{\varepsilon_2} A_2}{\mathbf{coerce}\, M_1\ \mathbf{to}\ x_1 : A_1.\ \mathbf{coerce}\, M_2\ \mathbf{to}\ x_2 : A_2.\, \mathbf{return}_{\varepsilon} x_1 = \mathbf{coerce}\, M_2\ \mathbf{to}\ x_2 : A_2.\ \mathbf{coerce}\, M_1\ \mathbf{to}\ x_1 : A_1.\, \mathbf{return}_{\varepsilon} x_1}$	$\psi_{\varepsilon} \circ (m^{\varepsilon_1} \times m^{\varepsilon_2}) \circ (\mathrm{id} \times \eta^{\varepsilon_2}_{\mathbb{1}}) = \tilde{\psi}_{\varepsilon} \circ (m^{\varepsilon_1} \times m^{\varepsilon_2}) \circ (\mathrm{id} \times \eta^{\varepsilon_2}_{\mathbb{1}})$	For all ε-terms $t = \mathfrak{T}_1(t')$, $s = \mathfrak{T}_2(s')$: $t(s(\mathbf{x}_1, \ldots, \mathbf{x}_1), \ldots, s(\mathbf{x}_n, \ldots, \mathbf{x}_n)) = s(t(\mathbf{x}_1, \ldots, \mathbf{x}_n), \ldots, t(\mathbf{x}_1, \ldots, \mathbf{x}_n))$	when $\mathcal{T}_{\varepsilon_2}$ is affine, e.g.: read-only state and convex, upper and lower semilattices.
Isolated Swap	$\dfrac{\Gamma \vdash_{\varepsilon_1} M_1 : \mathbf{F}_{\varepsilon_1} A_1 \quad \Gamma \vdash_{\varepsilon_2} M_2 : \mathbf{F}_{\varepsilon_2} A_2 \quad \Gamma \vdash_{\varepsilon'} N}{\text{(same as Swap)}}$	$\dfrac{\Gamma \vdash_{\varepsilon_1} M_1 : \mathbf{F}_{\varepsilon_1} A_1 \quad \Gamma \vdash_{\varepsilon_2} M_2 : \mathbf{F}_{\varepsilon_2} A_2}{\mathbf{coerce}\, M_1\ \mathbf{to}\ x_1 : A_1.\ \mathbf{coerce}\, M_2\ \mathbf{to}\ x_2 : A_2.\, \mathbf{return}_{\varepsilon} \star = \mathbf{coerce}\, M_2\ \mathbf{to}\ x_2 : A_2.\ \mathbf{coerce}\, M_1\ \mathbf{to}\ x_1 : A_1.\, \mathbf{return}_{\varepsilon} \star}$	$\psi_{\varepsilon} \circ (m^{\varepsilon_1} \times m^{\varepsilon_2}) \circ (\eta^{\varepsilon_1}_{\mathbb{1}} \times \eta^{\varepsilon_2}_{\mathbb{1}}) = \tilde{\psi}_{\varepsilon} \circ (m^{\varepsilon_1} \times m^{\varepsilon_2}) \circ (\eta^{\varepsilon_1}_{\mathbb{1}} \times \eta^{\varepsilon_2}_{\mathbb{1}})$	For all ε-terms $t = \mathfrak{T}_1(t')$, $s = \mathfrak{T}_2(s')$: $t(s(\mathbf{x}, \ldots, \mathbf{x}), \ldots, s(\mathbf{x}, \ldots, \mathbf{x})) = s(t(\mathbf{x}, \ldots, \mathbf{x}), \ldots, t(\mathbf{x}, \ldots, \mathbf{x}))$	when $\mathcal{T}\varepsilon_1$ is affine: read-only state and convex, upper and lower semilattices.
Unique	B $\dfrac{\Gamma \vdash_{\varepsilon} M_i : \mathbf{F}_{\varepsilon} \mathbf{0}, i = 1, 2}{M_1 = M_2}$	(same as utilitarian form)	$F_{\varepsilon} \mathbb{0} = \mathbb{0}, \mathbb{1}$	$\mathcal{T}_{\varepsilon}$ equates all ε-constants	all three state theories, all three semilattice theories, a single unparameterised exception, lifting
Pure Hoist	B F $\dfrac{\Gamma \vdash_{\varepsilon} M : \mathbf{F}_{\varepsilon} A \quad \Gamma, x : A \vdash_{\varepsilon'} N : \underline{B}}{\mathbf{return}_{\varepsilon}\ \mathbf{thunk}\ (\mathbf{coerce}\, M\ \mathbf{to}\ x : A.\, N) = M\ \mathbf{to}\ x : A.\, \mathbf{return}_{\varepsilon}\ \mathbf{thunk}\ N}$	B F $\dfrac{\Gamma \vdash_{\varepsilon} M : \mathbf{F}_{\varepsilon} A}{\mathbf{return}_{\varepsilon}\ \mathbf{thunk}\ M = M\ \mathbf{to}\ x : A.\, \mathbf{return}_{\varepsilon}\ \mathbf{thunk}\ \mathbf{return}_{\varepsilon} x}$	F $L^{\varepsilon} \eta^{\varepsilon}_{W} = \eta^{\varepsilon}_{\lvert F_{\varepsilon} W \rvert}$	all ε-terms are equal to variables in $\mathcal{T}_{\varepsilon}$	the empty theory, inconsistent theories
Hoist	B F $\dfrac{\Gamma \vdash_{\varepsilon} M : \mathbf{F}_{\varepsilon} A \quad \Gamma, x : A \vdash_{\varepsilon'} N : \underline{B}}{M\ \mathbf{to}\ x : A.\ \mathbf{return}_{\varepsilon}\ \mathbf{thunk}\ (\mathbf{coerce}\, M\ \mathbf{to}\ x : A.\, N) = M\ \mathbf{to}\ x : A.\, \mathbf{return}_{\varepsilon}\ \mathbf{thunk}\ N}$	B F $\dfrac{\Gamma \vdash_{\varepsilon} M : \mathbf{F}_{\varepsilon} A}{M\ \mathbf{to}\ x : A.\ \mathbf{thunk}\ \mathbf{return}_{\varepsilon}(x, \mathbf{thunk}\ M) = M\ \mathbf{to}\ x : A.\ \mathbf{thunk}\ \mathbf{return}_{\varepsilon}(x, \mathbf{thunk}\ \mathbf{return}_{\varepsilon} x)}$	F $L^{\varepsilon} \langle \eta^{\varepsilon}, \mathrm{id} \rangle = \mathrm{str}^{\varepsilon} \circ \delta$	all ε-terms are either a variable or independent of their variables via $\mathcal{T}_{\varepsilon}$	all theories containing only constants: lifting and exceptions

Figure 7. Abstract Optimisations

9

2011/11/16

Further reading

Teasers

Details in the paper, and:

- An extended example:

$$\begin{aligned}\text{Exceptions} + (\text{Read Only} \otimes \text{Write Only} \otimes \text{Read-Write} \otimes \\ (\text{Exceptions} + \text{Input} + \text{Output} + \\ (\text{Non-determinism} \otimes \text{Lifting})))\end{aligned}$$

  ($2^9 = 512$ effect sets).
- Modular validation of optimisations.
- More expressible language (recursion + CBPV).
- More optimisations.
- Further work.

Summary

Conclusions

- This work unified and generalised existing work: a step towards a science and an engineering discipline.
- The algebraic approach is fruitful: clarifies and unveils both connections and constructions.
- Category theory was crucial to our formulation and for forming the connections between the different areas that were unified.

Some further work

- Effect reconstruction
- Handlers
- Automation
- More effects
- Locality
- Concurrency
- Better program logics (Hoare, modal, etc.).