Algebraic Foundations for Type and Effect Analysis

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

Gordon Plotkin
<gdp@ed.ac.uk>

EWCE, March 18, 2011

(another work in progress)

Laboratory for Foundations of Computer Science

THE UNIVERSITY of EDINBURGH informatics

Problem

Effect Systems

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

Effect-Dependent Optimisations [Benton et al.]

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

Difficulty

Change language $\implies$ reprove from scratch.

Plan

Solution

General semantic account of effect type systems.

Tool

Algebraic theory of effects.

Strategy

Observation [Wadler]

Change notation:

$$\Gamma \vdash M : T_{\varepsilon} A$$

T_{ε} behaves like a monad.

Our Idea

- Elements in ε are effect operations.
- ε is the signature for T_{ε}.
- Which equations?

Bird's Eye

Call-by-push-value (CBPV)

$\Downarrow$

Multi-adjunctive intermediate language (MAIL)

$\Downarrow$

Semantics

$\Downarrow$

Optimisations (logic)

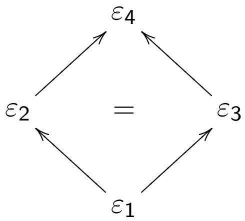
Talk Structure

- Abstract, non-algebraic, view of theory.
  - MAIL
  - Semantics.
  - Digression: generalised handlers.
- Algebraic instantiation.
  - Algebraic MAIL
  - Semantics.
  - Conservative restriction models.
  - Calculating conservative restrictions.
- Optimisations
- Modular approximation model.
- Conclusions.

Post Analysis

Effect Hierarchy

Category $\mathcal{E}$ with ε as objects. Typically partial order of effect sets.



Syntax

$\mathcal{E}$-MAIL

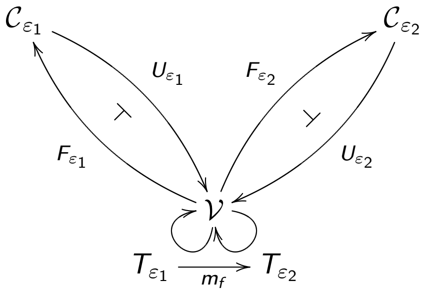
Multiple computation kinds $\texttt{Comp}_\varepsilon$, type constructors F_ε and U_ε.
Explicit coercion along $\mathcal{E}$ morphisms:

$$\frac{\Gamma \vdash_{\varepsilon_1} M : F_{\varepsilon_1} A}{\Gamma \vdash_{\varepsilon_2} \texttt{coerce}_{f:\varepsilon_1 \to \varepsilon_2} M : F_{\varepsilon_2} A}$$

Laboratory for Foundations of Computer Science

THE UNIVERSITY of EDINBURGH informatics

Semantics



$\mathcal{C}_{\varepsilon_1}$
$\mathcal{C}_{\varepsilon_2}$
U_{ε_1}
F_{ε_2}
$\top$
$\perp$
F_{ε_1}
U_{ε_2}
$\mathcal{V}$
T_{ε_1}
m_f
T_{ε_2}

Digression: Non-Algebraic Handlers

General Exceptional Syntax

$$\frac{\Gamma \vdash_{\varepsilon_1} M : F_{\varepsilon_1} A_1 \quad \vdash_h H : (A_9; \varepsilon_1) \Rightarrow (\underline{B}; \varepsilon_2) \quad \Gamma \vdash_v P : A_9 \quad \Gamma, x : A_1 \vdash_{\varepsilon_2} N : A_9 \to \underline{B}}{\Gamma \vdash_{\varepsilon_2} \mathtt{try}\ M\ \mathtt{with}\ H\ \mathtt{@}\ P\ \mathtt{as}\ x\ \mathtt{in}\ N : \underline{B}}$$

Semantics

$[\![\underline{B}]\!] \in \mathcal{C}_{\varepsilon_2}$, $[\![H]\!] \in \mathcal{C}_{\varepsilon_1}$:

$$U_{\varepsilon_2}([\![\underline{B}]\!]^{[\![A_9]\!]}) \cong U_{\varepsilon_1}[\![H]\!]$$

Digression: Non-Algebraic Handlers

General Exceptional Syntax

$$\frac{\Gamma \vdash_{\varepsilon_1} M : F_{\varepsilon_1} A_1 \quad \vdash_h H : (A_9; \varepsilon_1) \Rightarrow (\underline{B}; \varepsilon_2) \qquad \Gamma \vdash_v P : A_9 \quad \Gamma, x : A_1 \vdash_{\varepsilon_2} N : A_9 \to \underline{B}}{\Gamma \vdash_{\varepsilon_2} \mathtt{try}\ M\ \mathtt{with}\ H\ \mathtt{@}\ P\ \mathtt{as}\ x\ \mathtt{in}\ N : \underline{B}}$$

Semantics

$[\![\underline{B}]\!] \in \mathcal{C}_{\varepsilon_2}$, $[\![H]\!] \in \mathcal{C}_{\varepsilon_1}$:

$$U_{\varepsilon_2}([\![\underline{B}]\!]^{[\![A_9]\!]}) \cong U_{\varepsilon_1}[\![H]\!]$$

Examples

- Exception handlers.
- Logging.
- Effect reification.

Integrated Effects

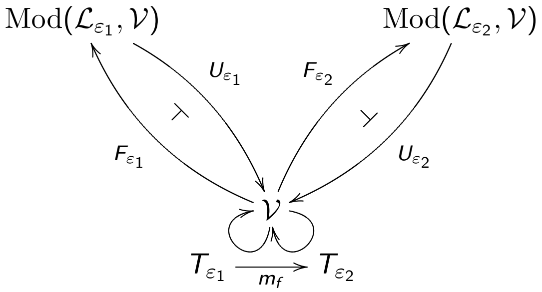
Overall effect equational theory $\mathcal{L} = \langle \Sigma, E \rangle$.
Effect hierarchy $\mathcal{E}$ a partial order of subsets $\varepsilon \subseteq \Sigma$.

$$\frac{\Gamma \vdash_v V : A_2 \quad \Gamma \vdash_\varepsilon M : A_1 \to \underline{B}}{\Gamma \vdash_\varepsilon \mathsf{op}_V M : \underline{B}} \ \mathsf{op} : A_1 \to A_2 \in \varepsilon$$

Semantics

$$\begin{array}{ccc}
\Sigma_{\varepsilon_1} & \xhookrightarrow{\Sigma_{\varepsilon_1 \subseteq \varepsilon_2}} & \Sigma_{\varepsilon_2} \\
\downarrow & = & \downarrow \\
\mathcal{L}_{\varepsilon_1} & \xrightarrow[G_{\varepsilon_1 \subseteq \varepsilon_2}]{} & \mathcal{L}_{\varepsilon_2}
\end{array}$$

Derived Semantics



Models

Benchmark Model

$$\mathcal{L}_{\varepsilon} := \mathcal{L}$$

- 👍 Original meaning.
- 👎 Discards effect analysis.

Relating Models

Logical relations for comparing other models against benchmark.

Conservative Restriction

$\mathcal{L}_\varepsilon$:= all terms over ε, and $\mathcal{L}$ equations between them.

Categorically:

$$\begin{array}{ccc} \Sigma_\varepsilon & \hookrightarrow & \Sigma \\ {\scriptstyle \text{full}}\downarrow & = & \downarrow \\ \mathcal{L}_\varepsilon & \underset{\text{faithful}}{\hookrightarrow} & \mathcal{L} \end{array}$$

- Original meaning.
- Uses effect analysis.
- Finding $\mathcal{L}_\varepsilon$ is non-trivial.

Modularity

Idea

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

Sum Theorem

For consistent finitary Lawvere theories:

$$(\mathcal{L}^1 + \mathcal{L}^2)_{\varepsilon_1 + \varepsilon_2} = \mathcal{L}^1_{\varepsilon_1} + \mathcal{L}^2_{\varepsilon_2}$$

Tensor Counterexample

Eckmann-Hilton:

$$(\text{Monoids} \otimes \text{Monoids})_{\{\cdot, 1\} + \emptyset} = \text{Commutative Monoids}$$

In particular: $x \cdot y = y \cdot x$

lfcs Laboratory for Foundations of Computer Science

THE UNIVERSITY of EDINBURGH informatics

Pragmatic Modularity

Observation

In practice:

- Sum: Free theories.
- Tensor: Global state, reader and writer.

Our Idea

Analyse restrictions in these cases only.

- 👍 Works in **Set**.
- 👎 Subtle in ω**CPO**: Works for the above theories, but may fail for others (non-determinism).

Optimisations

Validity

$\mathcal{M} \models M = N \iff [\![M]\!] = [\![N]\!]$ in $\mathcal{M}$.

Cataloguing Optimisations

For existing transformations:

- Validate.
- Classify.
- Generalise.

Optimisation Taxonomy

Structural

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

$$\texttt{coerce}_f(\texttt{coerce}_g M) = \texttt{coerce}_{f \circ g} M$$

- Handlers:
  - β: $\texttt{try}\ (\texttt{return}_\varepsilon\ V)\ \texttt{with}\ H\ \texttt{@}\ P\ \texttt{as}\ x\ \texttt{in}\ N = N[V/x]`P$ (Pretnar's thesis)
  - More can be said for user-defined handlers.
  - Others?

Algebraic

Equations in the underlying Lawvere theory, e.g.:

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

Optimisation Taxonomy

Abstract

Monadic properties of $\mathcal{C}_\varepsilon$ yield optimisations, e.g. discard:

$$\frac{T_\varepsilon \text{ affine} \quad \Gamma \vdash_\varepsilon M : F_\varepsilon A \quad \Gamma \vdash_{\varepsilon'} N : B}{\Gamma \vdash_{\varepsilon'} (\mathtt{coerce}_f M) \ \mathtt{to}\ x \ \mathtt{in}\ N = N}$$

Algebraic View

When T_ε is algebraic:

$$T_\varepsilon \text{ affine} \iff \begin{array}{c} f \\ / \quad \backslash \\ x \ \cdots \ x \end{array} = x \qquad \text{(absorption law holds)}$$

- Assists recognition.
- Modularity of combination.

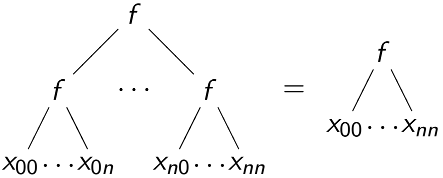
More [Abstract Optimisations]

Copy Optimisation

$$\frac{T_\varepsilon \text{ relevant} \quad \Gamma \vdash_\varepsilon M : F_\varepsilon A \quad \Gamma, x : A, y : A \vdash_{\varepsilon'} N : \underline{B}}{\begin{aligned}(\mathtt{coerce}_f\, M)\ \mathtt{to}\ x\ \mathtt{in}\ (\mathtt{coerce}_f\, M)\ \mathtt{to}\ y\ \mathtt{in}\ N = \\ (\mathtt{coerce}_f\, M)\ \mathtt{to}\ x\ \mathtt{in}\ N[x/y]\end{aligned}}$$

Algebraic View

Relevant $\iff$

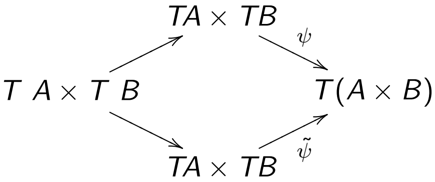


(idempotency)

Commutativity

Definition

Commutative monad T:



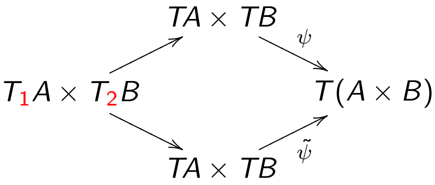
lfcs Laboratory for Foundations of Computer Science

THE UNIVERSITY of EDINBURGH informatics

Commutativity

Definition

Commuting monad morphisms $T_1 \to T \leftarrow T_2$:



More [Abstract Optimisations]

Commute Optimisation

$$T_{\varepsilon_1} \xrightarrow{m_1} T_\varepsilon \xleftarrow{m_2} T_{\varepsilon_2} \text{ commute}$$

$$\frac{\Gamma \vdash_{\varepsilon_i} M_i : F_{\varepsilon_i} A_i \quad \Gamma, x_1 : A_1, x_2 : A_2 \vdash_\varepsilon N : \underline{B}}{\begin{array}{r}(\mathtt{coerce}_{f_1} M_1) \mathtt{\ to\ } x_1 \mathtt{\ in\ } (\mathtt{coerce}_{f_2} M_2) \mathtt{\ to\ } x_2 \mathtt{\ in\ } N = \\ (\mathtt{coerce}_{f_2} M_2) \mathtt{\ to\ } x_2 \mathtt{\ in\ } (\mathtt{coerce}_{f_1} M_1) \mathtt{\ to\ } x_1 \mathtt{\ in\ } N\end{array}}$$

Algebraic View

m_1 translations commute with m_2 translations.

Corollary

$T_1 \to T_1 \otimes T_2 \leftarrow T_2$ commute.

More [Abstract Optimisations]

Hoist Optimisation

$$T_e = \mathrm{id}$$

$$\frac{\Gamma \vdash_\varepsilon M : F_\varepsilon A \quad \Gamma, x : A \vdash_{\varepsilon'} N : \underline{B}}{\begin{array}{r}\mathtt{return}_\varepsilon\,(\mathtt{thunk}_{\varepsilon'}((\mathtt{coerce}_f M)\ \mathtt{to}\ x\ \mathtt{in}\ N)) = \\ M\ \mathtt{to}\ x\ \mathtt{in}\ (\mathtt{return}_\varepsilon\,(\mathtt{thunk}_{\varepsilon'} N))\end{array}}$$

Caveats

- Generalise?
- Algebraic view?

Modular Approximation Model

e.g.:

$$\mathcal{L} := \mathcal{L}^1 + (\mathcal{L}^2 \otimes \mathcal{L}^3)$$

with $\mathcal{L}^i \sim \langle \Sigma_i, E_i \rangle$.
Approximation model, with $\varepsilon^i_1 + \varepsilon^i_2 + \varepsilon^i_3 \subseteq \Sigma^1 + \Sigma^2 + \Sigma^3$:

$$\begin{array}{ccccc}
 & & \mathcal{L}^1_{\varepsilon^4_1} + (\mathcal{L}^2_{\varepsilon^4_2} \otimes \mathcal{L}^3_{\varepsilon^4_3}) & & \\
 & \nearrow & & \nwarrow & \\
\mathcal{L}^1_{\varepsilon^2_1} + (\mathcal{L}^2_{\varepsilon^2_2} \otimes \mathcal{L}^3_{\varepsilon^2_3}) & & = & & \mathcal{L}^1_{\varepsilon^3_1} + (\mathcal{L}^2_{\varepsilon^3_2} \otimes \mathcal{L}^3_{\varepsilon^3_3}) \\
 & \nwarrow & & \nearrow & \\
 & & \mathcal{L}^1_{\varepsilon^1_1} + (\mathcal{L}^2_{\varepsilon^1_2} \otimes \mathcal{L}^3_{\varepsilon^1_3}) & &
\end{array}$$

Relating Models

$$\text{Approximation Model} \models M = N \implies \text{Benchmark Model} \models M = N$$

- 👍 Modularity.
- 👍 Equational soundness.
- 👎 Approximation only.

Summary

Conclusions

- Algebraically directed research.
- Modular account.
- A general account, in ascending degree.

Further work

- Effect reconstruction.
- More ω**CPO**.
- Logical relations.
- Handlers.
- Rewriting.
- More optimisations?
- Atkey's permissions.
- More effects.
- Concurrency.
- Presheaf models.