



Factorisation systems for logical relations  
and monadic lifting in  
type-and-effect system semantics

Ohad Kammar  
University of Oxford

Dylan McDermott  
University of Cambridge

# Type-and-effect systems

effect  type 

▶ Caching:  $\Gamma \vdash_{\{\text{write}\}} M : \text{int}$

$$(M, M) \equiv \text{let } x = M \text{ in } (x, x)$$

▶ Swapping:  $\Gamma \vdash_{\{\text{read}\}} M : 1, \Gamma \vdash_{\{\text{read}\}} K : 1$

$$M; K \equiv K; M$$

 semantic justification?

# Type-and-effect systems

signature  $\Sigma = \{\text{read} : 1 \rightarrow V, \text{write} : V \rightarrow 1, \text{raise} : \text{exn} \rightarrow 0, \dots\}$

parameter type  $V$

arity  $1$

ground types  $1, 0$

operation name  $\text{read}, \text{write}, \text{raise}$

$M, N ::= c \mid x \mid () \mid (M, N) \mid \text{fst } M \mid \text{snd } M \mid \text{elim}_0 M$   
|  $\text{inl } M \mid \text{inr } M \mid \text{match } M \text{ with } \{\text{inl } x. N_1, \text{inr } y. N_2\}$   
|  $\lambda x. M \mid MN \mid \text{let } x = M \text{ in } N$   
|  $\text{op } M$   $\text{op} \in \Sigma$

$A, B ::= b \mid 1 \mid A \times B \mid 0 \mid A + B$   
|  $A \xrightarrow{\varepsilon} B$   $\varepsilon \subseteq_{\text{fin}} \Sigma$

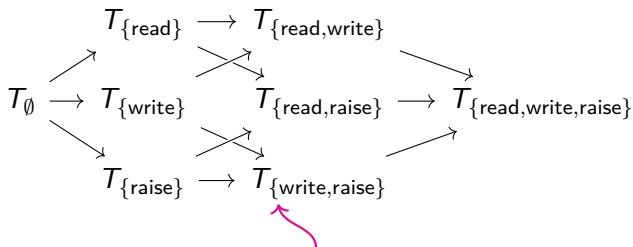
Erasure:

$\underline{\text{int}} \xrightarrow{\{\text{write}\}} 1 := \text{int} \rightarrow 1$        $\underline{x_1 : A_1, \dots, x_n : A_n} := x_1 : \underline{A_1}, \dots, x_n : \underline{A_n}$

$\Gamma \vdash_{\varepsilon} M : A$   $\Rightarrow$   $\underline{\Gamma} \vdash M : \underline{A}$

refined judgment unrefined judgment

# Type-and-effect systems



Refined models:

- ▶  $\varepsilon$ -monad  $T_{\varepsilon}$  for each  $\varepsilon \subseteq \Sigma$
- ▶  $\varepsilon$ -monad morphism  $T_{\varepsilon} \rightarrow T_{\varepsilon'}$  for each  $\varepsilon \subseteq \varepsilon'$

strong monad

$[[op]] : [[G]] \rightarrow T_{\varepsilon} [[G']]$

for each

$(op : G \rightarrow G') \in \varepsilon$

$[[\Gamma \vdash_{\varepsilon} M : A]] : [[\Gamma]] \rightarrow T_{\varepsilon} [[A]]$

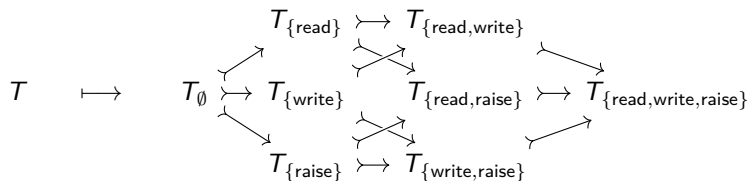
## Type-and-effect systems

$$\begin{aligned} & \left[ \left[ f: 1 \xrightarrow{\{\text{write}\}} \text{int} \vdash_{\{\text{write}\}} \quad (f(), f()) : \text{int} \times \text{int} \right] \right] \\ = & \left[ \left[ f: 1 \xrightarrow{\{\text{write}\}} \text{int} \vdash_{\{\text{write}\}} \text{let } x = f() \text{ in } (x, x) : \text{int} \times \text{int} \right] \right] \end{aligned}$$

but

$$\begin{aligned} & \left[ \left[ f: 1 \rightarrow \text{int} \vdash \quad (f(), f()) : \text{int} \times \text{int} \right] \right] \\ \neq & \left[ \left[ f: 1 \rightarrow \text{int} \vdash \text{let } x = f() \text{ in } (x, x) : \text{int} \times \text{int} \right] \right] \end{aligned}$$

# Constructing refined semantics



## Contribution: construction of refined semantics

- ▶  $T \mapsto T_\varepsilon$  via monad factorisation
- ▶ Sound and complete:

$$\llbracket \Gamma \vdash_\varepsilon M : G \rrbracket = \llbracket \Gamma \vdash_\varepsilon K : G \rrbracket \Leftrightarrow \llbracket \Gamma \vdash M : G \rrbracket = \llbracket \Gamma \vdash K : G \rrbracket$$

ground context

ground type

# Factoring monad morphisms

A factorisation system consists of:

- ▶ A class  $\mathcal{E}$  of morphisms  $e : X \twoheadrightarrow Y$
- ▶ A class  $\mathcal{M}$  of morphisms  $n : X \rightrightarrows Y$

“epis”

“monos”

such that:

- ▶ Every morphism can be factored:

$$\begin{array}{ccc}
 X & \xrightarrow{f} & Y \\
 e \searrow & = & \nearrow n \\
 & Z &
 \end{array}$$

- ▶  $\mathcal{E}$ ,  $\mathcal{M}$  closed under composition, contain isos
- ▶  $\mathcal{E}$  is left orthogonal to  $\mathcal{M}$ :

$$\begin{array}{ccc}
 W & \xrightarrow{e} \twoheadrightarrow & X \\
 \downarrow & = & \downarrow \\
 Y & \xrightarrow{n} \rightrightarrows & Z
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{ccc}
 W & \xrightarrow{e} \twoheadrightarrow & X \\
 \downarrow & = & \downarrow \\
 Y & \xrightarrow{n} \rightrightarrows & Z
 \end{array}$$

# Factoring monad morphisms

## Examples of factorisation systems

▶ **Set:**

(surjection, injection)

▶  $\omega$ **Cpo:**

(dense functions, full functions)

$$nx \leq ny \Rightarrow x \leq y$$

$$\omega\text{-chain-closure}(e[\text{domain}]) = \text{codomain}$$

▶ Functor categories:

(componentwise  $\mathcal{E}$ , componentwise  $\mathcal{M}$ )



# Factoring monad morphisms

## Theorem

$F$ -monad morphisms  $m : S \rightarrow T$  factor componentwise:

$$\begin{array}{ccc}
 SX & \xrightarrow{m_X} & TX \\
 \searrow e_X & \begin{array}{c} = \\ \swarrow n_X \end{array} & \nearrow n_X \\
 & RX &
 \end{array}$$

$e \in \mathcal{E} \implies Se, Fe \in \mathcal{E}$

If  $\mathcal{E}$  is closed under  $S$ , products, and  $F$  then:

▶  $R$  is an  $F$ -monad

▶  $e$  and  $n$  are  $F$ -monad morphisms

$$e_1, e_2 \in \mathcal{E} \implies e_1 \times e_2 \in \mathcal{E}$$

Pay as you go — drop:

▶  $F$ -algebra structure

▶ strength

▶ monad laws

$F$ -monad:

$$\begin{array}{ccccc}
 X \times F(TY) & \xrightarrow{\text{str}^F} & F(X \times TY) & \xrightarrow{F\text{str}^T} & F(T(X \times Y)) \\
 \text{id} \times \beta \downarrow & & = & & \downarrow \beta \\
 X \times TY & \xrightarrow{\text{str}^T} & & & T(X \times Y)
 \end{array}$$

# Canonical $S$ and $m$

Given  $\varepsilon \subseteq \Sigma$ , define

$$F_\varepsilon := \sum_{(\text{op}: G \rightarrow G') \in \varepsilon} [[G]] \times ([[G']] \Rightarrow (-))$$

$S_\varepsilon :=$  the free  $F_\varepsilon$ -monad

Apply factorisation theorem:

Assuming  $\mathcal{C}$  is:

- ▶ locally presentable; and
- ▶ bicartesian closed;

$$\begin{array}{ccc} S_\varepsilon X & \overset{\text{dashed}}{\longrightarrow} & TX \\ & \searrow & \nearrow \\ & T_\varepsilon X & \end{array}$$

=

$$T_{\varepsilon \subseteq \varepsilon'} : T_\varepsilon \rightarrow T_{\varepsilon'}$$

factorisation system  
functoriality

unrefined model

$\mapsto$

refined model

# Examples

In **Set**:

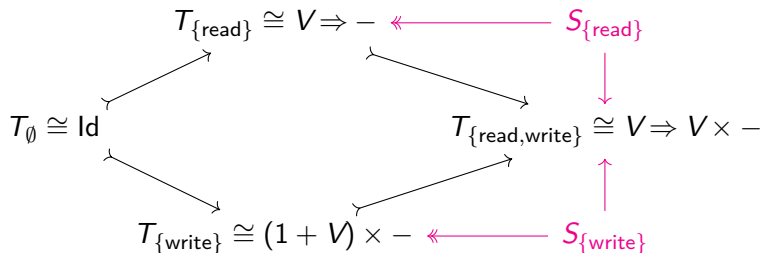
(for  $|V|, |R| > 1$ )

►  $T = V \Rightarrow V \times -$

(global state [Kammar'14])

$T = (V \Rightarrow - \Rightarrow R) \Rightarrow V \Rightarrow R$

(global state+continuations)



# Examples

In  $\omega\mathbf{Cpo}$ :

▶  $T = (- + E)_\perp$  (exceptions+partiality [Kammar-Plotkin'12])

$$T_{\{\text{diverge}\}} \cong (-)_\perp$$

# Examples

finite sets and injections

$[\mathbb{I}, \mathbf{Set}]$

►  $TXn = V^n \Rightarrow \int^{m \in \mathbb{I}} \mathbb{I}(n, m) \times V^m \times Xm$  (local state)

$$T_{\{\text{read}, \text{write}\}} Xn \cong V^n \Rightarrow V^n \times Xn$$

$T_{\{\text{read}, \text{write}, \text{alloc}\}}$

commutative monad

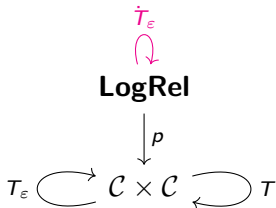
$$T_{\{\text{alloc}\}} Xn \cong \int^{m \in \mathbb{I}} \mathbb{I}(n, m) \times V^{m-n} \times Xm$$

## Goal (Completeness)

$$\overset{\text{ground}}{\llbracket \Gamma \vdash_{\varepsilon} M : G \rrbracket} = \llbracket \Gamma \vdash_{\varepsilon} N : G \rrbracket \quad \Leftrightarrow \quad \llbracket \Gamma \vdash M : G \rrbracket = \llbracket \Gamma \vdash N : G \rrbracket$$

Strategy:

- ▶ construct **LogRel**: category for logical relations
- ▶ lift  $(T_{\varepsilon}, T)$  to **LogRel**
- ▶ Interpret programs in refined **LogRel** model



predicates

## Definition (Fibration for logical relations [Katsumata '13])

For a bi-ccc  $\mathcal{C}$ :

*Fibration for logical relations*  $p : \mathcal{D} \rightarrow \mathcal{C}$ : faithful functor such that:

- ▶  $p$  is a bifibration;
- ▶  $\mathcal{D}$  bicartesian closed,  $p$  preserves bi-cc structure
- ▶ fibres have small products  $\wedge$

predicates have  
inverse and direct  
images

predicate  
conjunction

Use change-of-base:

binary  
logical  
relations

$$\begin{array}{ccc} \mathbf{LogRel} & \longrightarrow & \mathcal{D} \\ \downarrow & \lrcorner & \downarrow p \\ \mathcal{C} \times \mathcal{C} & \xrightarrow{\times} & \mathcal{C} \end{array}$$

## $\mathcal{M}$ -cod bifibration [Hughes and Jacobs '03]

In a factorisation system,  $\text{cod} : \mathcal{M} \rightarrow \mathcal{C}$  is a bifibration:

inverse images

over a category with pullbacks of  $\mathcal{M}$ -monos

$$\begin{array}{ccc} f^*P & \longrightarrow & P \\ f^*m \downarrow \lrcorner & & \downarrow m \\ X & \xrightarrow{f} & Y \end{array}$$

direct images

$$\begin{array}{ccc} P & \twoheadrightarrow & f_*P \\ m \downarrow & = & \downarrow m' \\ X & \xrightarrow{f} & Y \end{array}$$

## Definition (Factorisation systems for logical relations)

A factorisation system of a bi-ccc such that:

- ▶  $\mathcal{M}$  contains only monomorphisms
  - ▶  $\mathcal{M}$  closed under coproducts,  $\mathcal{E}$  under products
  - ▶ Fibres have small products
- $\text{cod}$  is faithful
- $p$  is bi-ccc



# Monadic lifting

$\Sigma$ -monad

fibration for  
logical relations

A  $\Sigma$ -lifting of  $T$  along  $p$  is a  $\Sigma$ -monad  $\dot{T}$  on  $\mathcal{D}$  above  $T$ :

$$\begin{array}{ccccc}
 X & \xrightarrow{\eta} & \dot{T}X & \xleftarrow{\mu} & \dot{T}(\dot{T}X) & & [\dot{G}] & \xrightarrow{[\dot{\text{op}}]} & \dot{T}[\dot{G}'] & & \mathcal{D} \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow p \\
 pX & \xrightarrow{\eta} & T(pX) & \xleftarrow{\mu} & T(T(pX)) & & [G] & \xrightarrow{[\text{op}]} & T[G'] & & \mathcal{C}
 \end{array}$$

for all  $(\text{op} : G \rightarrow G') \in \Sigma$ .

$$\begin{array}{c}
 \text{LogRel} \quad \begin{array}{c} \curvearrowright \dot{T} \end{array} \\
 \downarrow p \\
 \mathcal{C} \quad \curvearrowright T
 \end{array}$$

# Free lifting

Key idea [Plotkin and Power'03]

$$\begin{array}{ccc} G \rightarrow TG' & \longleftrightarrow & \alpha_X^{\text{op}} : (G' \Rightarrow TX) \rightarrow (G \Rightarrow TX) \\ \text{generic effect} & & \text{algebraic operation} \end{array}$$

$Y \in \mathcal{R}X$  when

and for all  $(\text{op} : G \rightarrow G') \in \Sigma$ :

$$\begin{array}{ccc} X & \xrightarrow{\dot{\eta}} & Y \\ \downarrow & & \downarrow \\ pX & \xrightarrow{\eta} & T(pX) \end{array}$$

$$\begin{array}{ccc} [\dot{G}] \Rightarrow Y & \xrightarrow{\dot{\alpha}_X^{\text{op}}} & [\dot{G}] \Rightarrow Y \\ \downarrow & & \downarrow \\ [G] \Rightarrow pX & \xrightarrow{\alpha_X^{\text{op}}} & [G] \Rightarrow T(pX) \end{array}$$

$$\dot{T}X := \bigwedge \mathcal{R}X$$

- ▶  $\dot{T}$  is a lifting of  $T$  to  $\mathcal{D}$
- ▶  $\dot{T}$  is initial: if  $\dot{T}'$  is a lifting then  $\dot{T} \leq \dot{T}'$

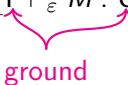
## Theorem (Soundness and Completeness)

If:

- ▶ Factorisation system for logical relations is well-powered
- ▶ Diagonals  $\delta : X \rightarrow X \times X$  are monos
- ▶  $\mathcal{C}$  is locally presentable bi-ccc

then

$$\llbracket \Gamma \vdash_{\varepsilon} M : G \rrbracket = \llbracket \Gamma \vdash_{\varepsilon} N : G \rrbracket \quad \Leftrightarrow \quad \llbracket \Gamma \vdash M : G \rrbracket = \llbracket \Gamma \vdash N : G \rrbracket$$

ground

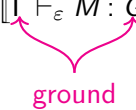
# Contributions

Given an unrefined model, with:

- ▶ A suitable category  $\mathcal{C}$  (factorisation system, etc.)
- ▶ **Any**  $\Sigma$ -monad on  $\mathcal{C}$

Can construct a refined model

- ▶ Containing simpler monads  $T_\varepsilon$
- ▶ With a completeness theorem:

$$\llbracket \Gamma \vdash_\varepsilon M : G \rrbracket = \llbracket \Gamma \vdash_\varepsilon N : G \rrbracket \iff \llbracket \Gamma \vdash M : G \rrbracket = \llbracket \Gamma \vdash N : G \rrbracket$$


ground