Logical relations, fibrations, and definability

Philip Saville, University of Oxford

(jww Ohad Kammar & Shin-ya Katsumata)

14th April 2022, Tallinn

$\rightsquigarrow$ loosely based on POPL '21 paper

Motivation

Aim: restrict the maps in a semantic model
to those satisfying some property

Motivation

Aim: restrict the maps in a semantic model to those satisfying some property

$\boxed{\text{starting model}} \rightsquigarrow \boxed{\text{model with only maps satisfying some predicate}}$

Motivation

eg System F: parametric maps
PCF: definable maps
(modulo approximation)

Aim: restrict the maps in a semantic model
to those satisfying some property

$\boxed{\text{starting model}} \rightsquigarrow \boxed{\text{model with only maps satisfying some predicate}}$

Motivation

eg System F: parametric maps
PCF: definable maps
(modulo approximation)

Aim: restrict the maps in a semantic model to those satisfying some property

for effectful CBV languages

$\boxed{\text{starting model}} \rightsquigarrow \boxed{\text{model with only maps satisfying some predicate}}$

Motivation

eg System F: parametric maps
PCF: definable maps
(modulo approximation)

Aim: restrict the maps in a semantic model to those satisfying some property

for effectful CBV languages

starting model $\rightsquigarrow$ model with only maps satisfying some predicate

Strategy: use fibrations, logical relations, and glueing

Three movements

1. Restricting models via fibrations (‘categories of concrete relations’)
2. Logical relations for effectful languages
3. Full completeness

   = building a model in which every map is definable

Three movements

1. Restricting models via fibrations (‘categories of concrete relations’)
2. Logical relations for effectful languages
3. Full completeness

= building a model in which every map is definable

related to full abstraction and completeness

exact link = work in progress!

Three movements

1. Restricting models via fibrations (‘categories of concrete relations’)

2. Logical relations for effectful languages

   + conjecture extension to an extrinsic, 2-categorical story

3. Full completeness

   = building a model in which every map is definable

related to full abstraction and completeness

exact link = work in progress!

Syntax and semantics of λ_{ml}

= Moggi's monadic metalanguage

Syntax of λ_{ml} = Moggi's monadic metalanguage

types: $\beta \in \mathrm{Base} \mid \sigma_1 \times \sigma_2 \mid \sigma \to \tau$

terms: $x \mid \xi \in \mathrm{Prim} \mid \mathrm{op} \in \mathrm{EfOp}$

$\mid M\, N \mid \lambda x\,.\,M \mid \pi_1(M) \mid \pi_2(M) \mid \langle M, M' \rangle \mid ()$

Syntax of λ_{ml} = Moggi's monadic metalanguage

types: $\beta \in \mathrm{Base} \mid \sigma_1 \times \sigma_2 \mid \sigma \to \tau \mid T\sigma$

terms: $x \mid \xi \in \mathrm{Prim} \mid \mathrm{op} \in \mathrm{EfOp}$

$\mid M\,N \mid \lambda x\,.\,M \mid \pi_1(M) \mid \pi_2(M) \mid \langle M, M' \rangle \mid ()$

$\mid \mathit{return}(M) \mid \mathrm{let}\ x = N\ \mathrm{in}\ M$

Semantics of λ_{ml}

CCC

interpretation of base types and constants

model = $(\mathcal{M}, T, s)$

strong monad

Semantics of λ_{ml}

CCC

model = $(\mathscr{M}, T, s)$

interpretation of base types and constants

strong monad

Interpretation of types:

- $s[\![\sigma]\!]$ as for simply-typed lambda calculus
- $s[\![T\sigma]\!] = T(s[\![\sigma]\!])$

Semantics of λ_{ml}

model = $(\mathscr{M}, T, s)$

CCC

strong monad

interpretation of base types and constants

Interpretation of types:

- $s[\![\sigma]\!]$ as for simply-typed lambda calculus
- $s[\![T\sigma]\!] = T(s[\![\sigma]\!])$

Interpretation of terms:

- $s[\![M : \sigma]\!]$ as for simply-typed lambda calculus
- $s[\![\mathrm{return}(M) : T\sigma]\!] = \eta \circ s[\![\sigma]\!]$
- $s[\![\mathrm{let}\ldots]\!]$ interpreted using monadic bind

1: Concrete relations (by example)

a flexible method for restricting models

Example: read-only state $(\mathcal{S}\mathrm{et}, T, s)$

syntax:

base types: bool

primitives: $\mathrm{tt} : \mathrm{bool}, \mathrm{ff} : \mathrm{bool}, \mathrm{or} : \mathrm{bool} \times \mathrm{bool} \to \mathrm{bool}$, etc

effect operations: $\mathrm{read} : 1 \to T(\mathrm{bool})$

Example: read-only state (Set, T, s)

syntax:

base types: bool

primitives: $\mathrm{tt} : \mathrm{bool}, \mathrm{ff} : \mathrm{bool}, \mathrm{or} : \mathrm{bool} \times \mathrm{bool} \to \mathrm{bool}$, etc

effect operations: $\mathrm{read} : 1 \to T(\mathrm{bool})$

semantics:

base category: Set

monad: reader $T(X) = (2 \Rightarrow X)$

computations of type σ
= set maps $s[\![\Gamma]\!] \to (2 \Rightarrow s[\![\sigma]\!])$

Example: read-only state (Set, T, s)

syntax:

base types: bool

primitives: $\mathrm{tt} : \mathrm{bool}, \mathrm{ff} : \mathrm{bool}, \mathrm{or} : \mathrm{bool} \times \mathrm{bool} \to \mathrm{bool}$, etc

effect operations: $\mathrm{read} : 1 \to T(\mathrm{bool})$

semantics:

base category: Set

monad: reader $T(X) = (2 \Rightarrow X)$

computations of type σ
= set maps $s[\![\Gamma]\!] \to (2 \Rightarrow s[\![\sigma]\!])$

interpretation of base types and constants:

$$s(\mathrm{bool}) = 2 = \{ \top , \bot \}$$
$$s(\mathrm{read}) = \lambda x . \lambda i . i : 1 \to (2 \Rightarrow 2) = T(s[\![\mathrm{bool}]\!])$$
$$s(\mathrm{bool}) = \lambda x . \top : 1 \to 2 = s[\![\mathrm{bool}]\!]$$

(Set, T, s) has too many maps (Matache & Staton)

base category: Set

monad: reader $T(X) = (2 \Rightarrow X)$

computations of type σ

$=$ set maps $s[\![\Gamma]\!] \to (2 \Rightarrow s[\![\sigma]\!])$

interpretation of base types and constants:

$$s(\mathrm{bool}) = 2$$

$$s(\mathrm{read}) = \lambda x\,.\,\lambda i\,.\,i : 1 \to (2 \Rightarrow 2) = T(s[\![\mathrm{bool}]\!])$$

$$s(\mathrm{bool}) = \lambda x\,.\,\top : 1 \to 2 = s[\![\mathrm{bool}]\!]\text{, etc}$$

$$\kappa : (1 \Rightarrow T2) \to T2$$

$$\kappa(g) = \begin{cases} \lambda i\,.\,\top & \text{if } g(\bullet) = \lambda i\,.\,\top \\ \lambda i\,.\,\bot & \text{else} \end{cases}$$

(Set, T, s) has too many maps (Matache & Staton)

base category: Set

monad: reader $T(X) = (2 \Rightarrow X)$

computations of type σ
= set maps $s[\![\Gamma]\!] \to (2 \Rightarrow s[\![\sigma]\!])$

interpretation of base types and constants:

$s(\mathrm{bool}) = 2$

$s(\mathrm{read}) = \lambda x\,.\,\lambda i\,.\,i : 1 \to (2 \Rightarrow 2) = T(s[\![\mathrm{bool}]\!])$

$s(\mathrm{tt}) = \lambda x\,.\,\top : 1 \to 2 = s[\![\mathrm{bool}]\!]$, etc

$$\kappa : (1 \Rightarrow T2) \to T2$$

$$\kappa(g) = \begin{cases} \lambda i\,.\,\top & \text{if } g(\bullet) = \lambda i\,.\,\top \\ \lambda i\,.\,\bot & \text{else} \end{cases}$$

behaves in a way no program can!

cf. parallel-or for PCF

(Set, T, s) has too many maps (Matache & Staton)

base category: Set

monad: reader $T(X) = (2 \Rightarrow X)$

computations of type σ
= set maps $s[\![\Gamma]\!] \to (2 \Rightarrow s[\![\sigma]\!])$

interpretation of base types and constants:

$$s(\mathrm{bool}) = 2$$

$$s(\mathrm{read}) = \lambda x \,.\, \lambda i \,.\, i : 1 \to (2 \Rightarrow 2) = T(s[\![\mathrm{bool}]\!])$$

$$s(\mathrm{tt}) = \lambda x \,.\, \top : 1 \to 2 = s[\![\mathrm{bool}]\!], \text{ etc}$$

$$\kappa : (1 \Rightarrow T2) \to T2$$

$$\kappa(g) = \begin{cases} \lambda i \,.\, \top & \text{if } g(\bullet) = \lambda i \,.\, \top \\ \lambda i \,.\, \bot & \text{else} \end{cases}$$

behaves in a way no program can!

cf. parallel-or for PCF

distinguishes contextually-equivalent programs:

$$\exists M, M' \,.\, \left(M \simeq_{ctx} M' \text{ but } [\![M]\!](\kappa) \neq [\![M']\!](\kappa)\right)$$

full abstraction fails

(Set, T, s) has too many maps (Matache & Staton)

base category: Set

monad: reader $T(X) = (2 \Rightarrow X)$

computations of type σ
= set maps $s[\![\Gamma]\!] \to (2 \Rightarrow s[\![\sigma]\!])$

interpretation of base types and constants:

$s(\mathrm{bool}) = 2$

$s(\mathrm{read}) = \lambda x\,.\,\lambda i\,.\,i : 1 \to (2 \Rightarrow 2) = T(s[\![\mathrm{bool}]\!])$

$s(\mathrm{tt}) = \lambda x\,.\,\top : 1 \to 2 = s[\![\mathrm{bool}]\!]$, etc

$$\kappa : (1 \Rightarrow T2) \to T2$$

$$\kappa(g) = \begin{cases} \lambda i\,.\,\top & \text{if } g(\bullet) = \lambda i\,.\,\top \\ \lambda i\,.\,\bot & \text{else} \end{cases}$$

$\longrightarrow$ behaves in a way no program can!

cf. parallel-or for PCF

distinguishes contextually-equivalent programs:

$$\exists M, M'\,.\,\big(M \simeq_{ctx} M' \text{ but } [\![M]\!](\kappa) \neq [\![M']\!](\kappa)\big)$$

$\longrightarrow$ full abstraction fails

κ is a bad map: want to remove it

Idea: restrict to maps preserving relations

Define a category $\mathbb{L}$ of 'predicates'

- objects:

  (X, R_0, R_1) with $X \in \mathrm{Set}$, $R_i \subseteq X^2$

- maps $(X, R_0, R_1) \to (Y, S_0, S_1)$:

  maps $f : X \to Y$ preserving the relation

Idea: restrict to maps preserving relations

Define a CCC $\mathbb{L}$ of 'predicates'

- objects: (X, R_0, R_1)
- maps $(X, R_0, R_1) \to (Y, S_0, S_1)$:

  maps $f : X \to Y$ preserving the relation

Idea: restrict to maps preserving relations

Define a CCC $\mathbb{L}$ of 'predicates'

- objects: (X, R_0, R_1)
- maps $(X, R_0, R_1) \to (Y, S_0, S_1)$:

  maps $f : X \to Y$ preserving the relation

- exponentials:

$$(X, R_0, R_1) \Rightarrow (Y, S_0, S_1) := \big(X \Rightarrow Y, R_0 \supset S_0, R_1 \supset S_1\big)$$

$$(f, g) \in (R_i \supset S_i) \iff \big((x, x')\ R_i \implies (f\,x, g\,x') \in S_i\big)$$

Idea: restrict to maps preserving relations

Define a CCC $\mathbb{L}$ of 'predicates'

- objects: (X, R_0, R_1)
- maps $(X, R_0, R_1) \to (Y, S_0, S_1)$:

  maps $f : X \to Y$ preserving the relation

- exponentials:

$$(X, R_0, R_1) \Rightarrow (Y, S_0, S_1) := \big(X \Rightarrow Y, R_0 \supset S_0, R_1 \supset S_1\big)$$

$$(f, g) \in (R_i \supset S_i) \iff \big((x, x')\ R_i \implies (f\,x, g\,x') \in S_i\big)$$

- terminal object:

$$(1, \top, \top)$$

$$\top = \{(\bullet, \bullet)\}$$

- products:

$$(X, R_0, R_1) \times (Y, S_0, S_1) = \big(X \times Y, R_0 \star S_0, R_1 \star S_1\big)$$

$\big((x_1, y_1), (x_2, y_2)\big) \in (R_i \star S_i) \iff (x_1, x_2) \in R_i$ and $(y_1, y_2) \in S_i$

Idea: restrict to maps preserving relations

Define a model $(\mathbb{L}, \hat{T}, \hat{s})$ **of 'predicates'**

- objects:
  (X, R_0, R_1) with $X \in \mathrm{Set}, R_i \subseteq X^2$
- maps $(X, R_0, R_1) \to (Y, S_0, S_1)$:
  maps $f : X \to Y$ preserving the relation

Idea: restrict to maps preserving relations

Define a model $(\mathbb{L}, \hat{T}, \hat{s})$ **of 'predicates'**

- objects:

  (X, R_0, R_1) with $X \in \mathrm{Set}$, $R_i \subseteq X^2$

- maps $(X, R_0, R_1) \to (Y, S_0, S_1)$:

  maps $f : X \to Y$ preserving the relation

- interpretation:

  $\hat{s}(\mathrm{bool}) = \big(2, \{(0,0), (1,1)\}, \{(0,0), (1,1)\}\big)$

Idea: restrict to maps preserving relations

Define a model $(\mathbb{L}, \hat{T}, \hat{s})$ **of ‘predicates’**

- objects:

  (X, R_0, R_1) with $X \in \mathrm{Set}$, $R_i \subseteq X^2$

- maps $(X, R_0, R_1) \to (Y, S_0, S_1)$:

  maps $f : X \to Y$ preserving the relation

- interpretation:

  $$\hat{s}(\mathrm{bool}) = \big(2, \{(\bot, \bot), (\top, \top)\}, \{(\bot, \bot), (\top, \top)\}\big)$$

- monad:

  $$\hat{T}(X, R_0, R_1) = \big(TX, (\hat{T}R)_0, (\hat{T}R)_1\big)$$

$$(h, h') \in (\hat{T}R)_i \iff (h\,i, h'\,i) \in R_i$$

Idea: restrict to maps preserving relations

Define a model $(\mathbb{L}, \hat{T}, \hat{s})$ **of 'predicates'**

- objects:

  (X, R_0, R_1) with $X \in \mathrm{Set}$, $R_i \subseteq X^2$

- maps $(X, R_0, R_1) \to (Y, S_0, S_1)$:

  maps $f : X \to Y$ preserving the relation

κ is not a map

$\kappa : (1 \Rightarrow \hat{T}\hat{s}[\![\mathrm{bool}]\!]) \to \hat{T}\hat{s}[\![\mathrm{bool}]\!]$

in $\mathbb{L}$

- interpretation:

  $$\hat{s}(\mathrm{bool}) = \big(2, \{(\bot, \bot), (\top, \top)\}, \{(\bot, \bot), (\top, \top)\}\big)$$

- monad:

  $$\hat{T}(X, R_0, R_1) = \big(TX, (\hat{T}R)_0, (\hat{T}R)_1\big)$$

  $(h, h') \in (\hat{T}R)_i \iff (h\,i, h'\,i) \in R_i$

Idea: restrict to maps preserving relations

…have a model $(\mathbb{L}, \hat{T}, \hat{s})$ of ‘predicates’ with

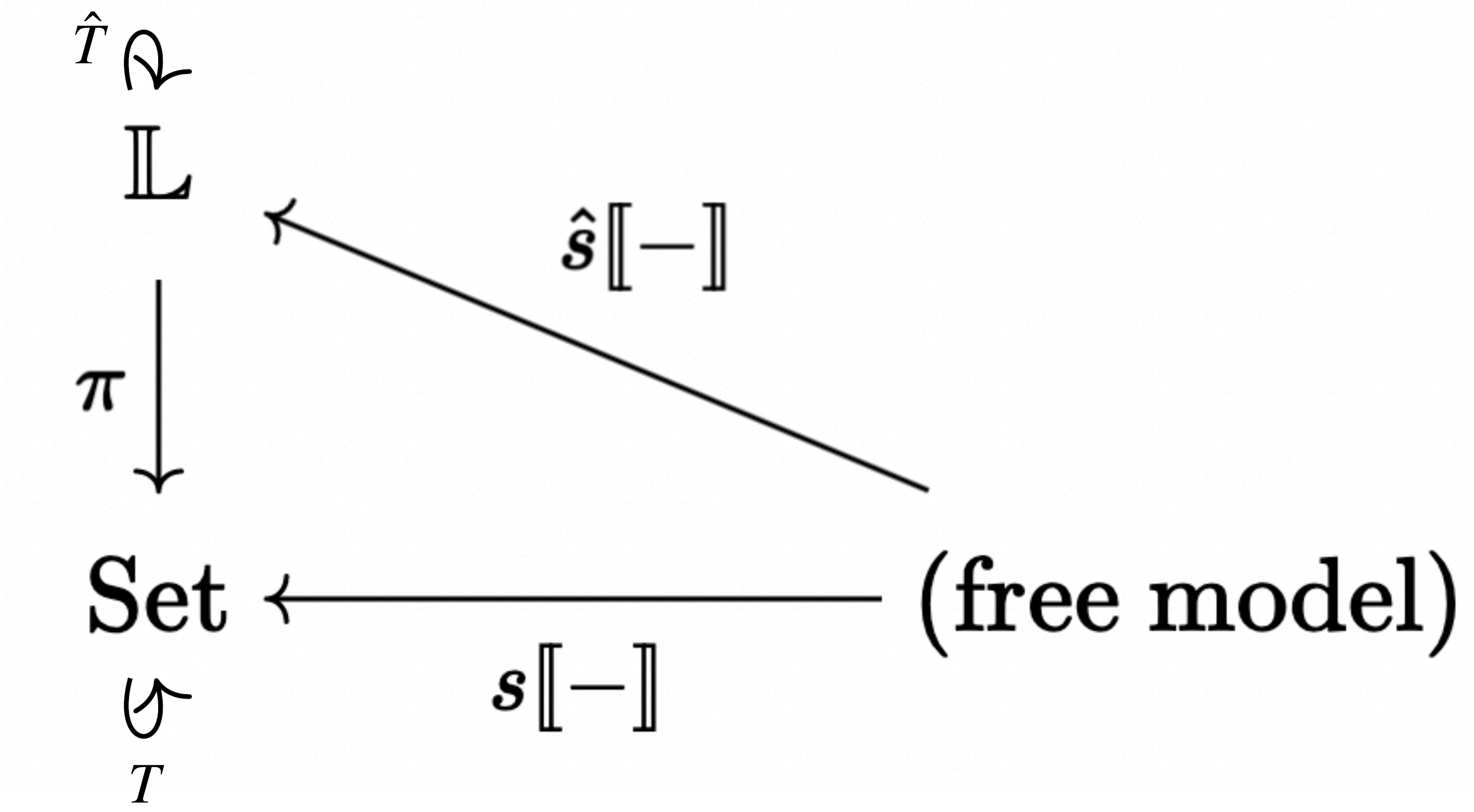
- morphism of models $\pi : \mathbb{L} \to \mathrm{Set}$:

  preserves

  ccc-structure

  monads

  semantic interpretation



Idea: restrict to maps preserving relations

…have a model $(\mathbb{L}, \hat{T}, \hat{s})$ of 'predicates' with

$$
\begin{tikzcd}
\mathbb{L} \arrow[loop above, "\hat{T}"] \arrow[d, "\pi"'] & \\
\mathrm{Set} \arrow[loop below, "T"] & (\text{free model}) \arrow[ul, "\hat{s}[\![-]\!]"'] \arrow[l, "s[\![-]\!]"]
\end{tikzcd}
$$

- morphism of models $\pi : \mathbb{L} \to \mathrm{Set}$:

    preserves

    - ccc-structure
    - monads
    - semantic interpretation

- new model refines the original one: $\mathbb{L}(\hat{s}[\![\sigma]\!], \hat{s}[\![\tau]\!]) \subseteq \mathrm{Set}(s[\![\sigma]\!], s[\![\tau]\!])$

Idea: restrict to maps preserving relations

…have a model $(\mathbb{L}, \hat{T}, \hat{s})$ of 'predicates' with

- morphism of models $\pi : \mathbb{L} \to \mathrm{Set}$:

  preserves

  ccc-structure

  monads

  semantic interpretation

problem: $\kappa \in s[\![(1 \to T\mathrm{bool}) \to T\mathrm{bool}]\!]$

$\implies \kappa \in \big(\text{carrier of } \hat{s}[\![(1 \to T\mathrm{bool}) \to T\mathrm{bool}]\!]\big)$

previous problem, internalised

- new model refines the original one: $\mathbb{L}(\hat{s}[\![\sigma]\!], \hat{s}[\![\tau]\!]) \subseteq \mathrm{Set}(s[\![\sigma]\!], s[\![\tau]\!])$

Idea: restrict to maps preserving relations

…have a model $(\mathbb{L}, \hat{T}, \hat{s})$ of 'predicates' with

- morphism of models $\pi : \mathbb{L} \to \mathrm{Set}$:

  preserves

  ccc-structure

  monads

  semantic interpretation

problem: $\kappa \in s[\![(1 \to T\mathrm{bool}) \to T\mathrm{bool}]\!]$

$\implies \kappa \in \big(\text{carrier of } \hat{s}[\![(1 \to T\mathrm{bool}) \to T\mathrm{bool}]\!]\big)$

previous problem, internalised

- new model refines the original one: $\mathbb{L}(\hat{s}[\![\sigma]\!], \hat{s}[\![\tau]\!]) \subseteq \mathrm{Set}(s[\![\sigma]\!], s[\![\tau]\!])$

$\boxed{\mathbb{L} \text{ removes } \kappa \text{ from the hom-set, but not the function space}}$

can still distinguish contextually-equivalent terms!

Concreteness: removing κ from the function space

Concreteness: removing κ from the function space

$\kappa \in \hat{s}[\![(1 \to T\mathrm{bool}) \to T\mathrm{bool}]\!]$ does not correspond to a global element:

there is no $g : 1 \to \hat{s}[\![(1 \to T\mathrm{bool}) \to T\mathrm{bool}]\!]$ in $\mathbb{L}$
such that $g(\bullet) = \kappa$

Concreteness: removing κ from the function space

$\kappa \in \hat{s}[\![(1 \to T\mathrm{bool}) \to T\mathrm{bool}]\!]$ does not correspond to a global element:

there is no $g : 1 \to \hat{s}[\![(1 \to T\mathrm{bool}) \to T\mathrm{bool}]\!]$ in $\mathbb{L}$
such that $g(\bullet) = \kappa$

so: restrict to objects (X, R_0, R_1) in which every $x \in X$
corresponds to a global element in $\mathbb{L}$

Concreteness: removing κ from the function space

$\kappa \in \hat{s}[\![(1 \to T\mathrm{bool}) \to T\mathrm{bool}]\!]$ does not correspond to a global element:

there is no $g : 1 \to \hat{s}[\![(1 \to T\mathrm{bool}) \to T\mathrm{bool}]\!]$ in $\mathbb{L}$
such that $g(\bullet) = \kappa$

so: restrict to objects (X, R_0, R_1) in which every $x \in X$
corresponds to a global element in $\mathbb{L}$

(X, R_0, R_1) is concrete if every $x : 1 \to X$ in Set lifts to $(1, \top, \top) \to (X, R_0, R_1)$

$x \in X \implies (x, x) \in R_i \ (i = 1,2)$

The subcategory of concrete objects

$$
\begin{tikzcd}
\mathrm{Conc}(\mathbb{L}) \arrow[r, "j"] & \mathbb{L} \arrow[loop above, "\hat{T}"] \arrow[d, "\pi"] & \\
& \mathrm{Set} \arrow[loop below, "T"] & \text{(free model)} \arrow[l, "s[\![-]\!]"] \arrow[ul, "\hat{s}[\![-]\!]"']
\end{tikzcd}
$$

interpretation of bool etc
restricts to $\mathrm{Conc}(\mathbb{L})$

The subcategory of concrete objects

$$
\begin{tikzcd}
\mathrm{Conc}(\mathbb{L}) \arrow[rr, "j"] & & \mathbb{L} \arrow[ll, bend right=40, "H"'] \arrow[ll, bend right=20, phantom, "\top"] \arrow[ll, bend left=20, phantom, "\top"] \arrow[ll, bend left=40, "K"] \arrow[d, "\pi"] \arrow[loop above, "\hat{T}"] & \\
& & \mathrm{Set} \arrow[loop below, "T"] & (\text{free model}) \arrow[l, "s[\![-]\!]"] \arrow[ul, "\hat{s}[\![-]\!]"']
\end{tikzcd}
$$

interpretation of bool etc
restricts to $\mathrm{Conc}(\mathbb{L})$

The subcategory of concrete objects

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
\mathrm{Conc}(\mathbb{L}) \arrow[r, "j" description] \arrow[r, phantom, shift left=3, "\top"] \arrow[r, phantom, shift right=3, "\top"] & \mathbb{L} \arrow[l, bend right=50, "H"'] \arrow[l, bend left=50, "K"] \arrow[d, "\pi"] \arrow[loop above, "\hat{T}"] & \\
& \mathrm{Set} \arrow[loop below, "T"] & (\text{free model}) \arrow[l, "\boldsymbol{s}[\![-]\!]"] \arrow[ul, "\hat{\boldsymbol{s}}[\![-]\!]"']
\end{tikzcd}
$$

H restricts (X, R_0, R_1) to those $x \in X$ such that $(x, x) \in R_i$ $(i = 1{,}2)$

interpretation of bool etc restricts to $\mathrm{Conc}(\mathbb{L})$

K expands (X, R_0, R_1):

$K(X, R_0, R_1) = (X, R_0 \cup \Delta, R_1 \cup \Delta)$

The subcategory of concrete objects

monad by
abstract
nonsense

H restricts (X, R_0, R_1) to
those $x \in X$ such that
$(x, x) \in R_i \ \ (i = 1,2)$

ccc by
abstract
nonsense:

$[\,-\Rightarrow=\,]_{\mathrm{Conc}} = H\big(j(\,-\,) \Rightarrow j(\,=\,)\big)$

$$\begin{tikzcd}
\mathbf{Conc}(\mathbb{L}) \arrow[loop above, "H\hat{T}j"] \arrow[rr, "j"] & & \mathbb{L} \arrow[loop above, "\hat{T}"] \arrow[ll, bend right=50, "H"'] \arrow[ll, bend left=50, "K"] \arrow[d, "\pi"] \\
 & & \mathbf{Set} \arrow[loop below, "T"] & \text{(free model)} \arrow[l, "s[\![-]\!]"] \arrow[ul, "\hat{s}[\![-]\!]"']
\end{tikzcd}$$

interpretation of bool etc
restricts to $\mathrm{Conc}(\mathbb{L})$

K expands (X, R_0, R_1):
$K(X, R_0, R_1) = (X, R_0 \cup \Delta, R_1 \cup \Delta)$

The subcategory of concrete objects

monad by
abstract
nonsense

H restricts (X, R_0, R_1) to
those $x \in X$ such that
$(x, x) \in R_i \ \ (i = 1{,}2)$

$$\begin{tikzcd}
\mathrm{Conc}(\mathbb{L}) \arrow[loop above, "H\hat{T}j"] \arrow[rr, "j"] & & \mathbb{L} \arrow[ll, bend right, "H"'] \arrow[ll, bend left, "K"] \arrow[loop above, "\hat{T}"] \arrow[d, "\pi"] & \\
& & \mathrm{Set} \arrow[loop below, "T"] & (\text{free model}) \arrow[l, "s[\![-]\!]"] \arrow[ul, "\hat{s}[\![-]\!]"']
\end{tikzcd}$$

ccc by
abstract
nonsense:

$[\,-\Rightarrow=\,]_{\mathrm{Conc}} = H\big(j(\,-\,)\Rightarrow j(\,=\,)\big)$

interpretation of bool etc
restricts to $\mathrm{Conc}(\mathbb{L})$

K expands (X, R_0, R_1):

$K(X, R_0, R_1) = (X, R_0 \cup \Delta, R_1 \cup \Delta)$

$\boxed{\text{key property: } \big[(X,\ldots)\Rightarrow(Y,\ldots)\big]_{\mathrm{Conc}} \cong \mathbb{L}\big(j(X,\ldots), j(Y,\ldots)\big) \subseteq Set(X,Y)}$

The subcategory of concrete objects

monad by
abstract
nonsense

ccc by
abstract
nonsense:

$[- \Rightarrow =]_{\mathrm{Conc}} = H\big(j(-) \Rightarrow j(=)$

K expands $(X, R$

$K(X, R_0, R_1) = (X, R_0 \cup$

$$
\begin{tikzcd}
\mathrm{Conc}(\mathbb{L}) \arrow[r, "j"] \arrow[r, phantom, shift left=4, "\top"] \arrow[r, phantom, shift right=4, "\top"] & \mathbb{L} \arrow[l, bend right, "H"'] \arrow[l, bend left, "K"] \arrow[d, "\pi"] & \\
& \mathrm{Set} & (\text{free model}) \arrow[l, "s[\![-]\!]"] \arrow[ul, "\hat{s}[\![-]\!]"']
\end{tikzcd}
$$

T

$, R_0, R_1)$ to
such that
$(i = 1,2)$

on of bool etc
to $\mathrm{Conc}(\mathbb{L})$

key property: $\big[(X, \ldots) \Rightarrow (Y, \ldots)\big]_{\mathrm{Conc}} \cong \mathbb{L}\big(j(X, \ldots), j(Y, \ldots)\big) \subseteq Set(X, Y)$

internalises the preservation condition

κ cannot be in function space!

Summing up

$$
\begin{tikzcd}
\mathrm{Set} \arrow[loop below, "T"] & (\text{free model}) \arrow[l, "\boldsymbol{s}[\![-]\!]"']
\end{tikzcd}
$$

κ in hom-sets and function spaces

Summing up

$$\begin{tikzcd}
\mathbb{L} \arrow[loop above, "\hat{T}"] \arrow[d, "\pi"] & \\
\mathrm{Set} \arrow[loop below, "T"] & (\text{free model}) \arrow[l, "s[\![-]\!]"] \arrow[ul, "\hat{s}[\![-]\!]"']
\end{tikzcd}$$

κ in function spaces

κ in hom-sets and function spaces

Summing up

$$
\begin{tikzcd}
\mathrm{Conc}(\mathbb{L}) \arrow[loop above, "H\hat{T}j"] \arrow[rr, "j"] \arrow[rr, phantom, bend left=25, "\top"] \arrow[rr, phantom, bend right=25, "\top"] & & \mathbb{L} \arrow[ll, bend right=40, "H"'] \arrow[ll, bend left=40, "K"] \arrow[loop above, "\hat{T}"] \arrow[d, "\pi"] & \\
& & \mathrm{Set} \arrow[loop below, "T"] & (\text{free model}) \arrow[l, "s[\![-]\!]"] \arrow[ul, "\hat{s}[\![-]\!]"']
\end{tikzcd}
$$

κ gone!

κ in function spaces

κ in hom-sets and function spaces

Abstracting away: categories of concrete relations

idea:

1. axiomatise relations by fibrations
2. ccc-structure via structured fibrations
3. monad defined using fibration
4. restrict to concrete objects

Abstracting away: categories of relations

1. axiomatise relations by fibrations

$$\begin{tikzcd} & \mathbb{E} \arrow[d, "p"] \\ \mathcal{M} \arrow[r, "F"'] & \mathbb{B} \end{tikzcd}$$

fibration

‘change-of-base’

Abstracting away: categories of relations

1. axiomatise relations by fibrations

objects: $(X, R) \in \mathscr{M} \times \mathbb{E}$ such that $FX = pR$

maps: $(f, \hat{f})$ in $\mathscr{M} \times \mathbb{E}$ such that $Ff = p(\hat{f})$

$$\begin{tikzcd}
\mathbb{K} \arrow[r] \arrow[d, "\pi"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\
\mathcal{M} \arrow[r, "F"'] & \mathbb{B}
\end{tikzcd}$$

fibration

‘change-of-base’

Abstracting away: categories of relations

1. axiomatise relations by fibrations

objects: $(X, R) \in \mathscr{M} \times \mathbb{E}$ such that $FX = pR$

maps: $(f, \hat{f})$ in $\mathscr{M} \times \mathbb{E}$ such that $Ff = p(\hat{f})$

fibration
$(X, R) \mapsto X$

$$\begin{tikzcd} \mathbb{K} \arrow[r] \arrow[d, "\pi"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\ \mathcal{M} \arrow[r, "F"'] & \mathbb{B} \end{tikzcd}$$

fibration

‘change-of-base’

Abstracting away: categories of relations

1. axiomatise relations by fibrations

objects: $(X, R) \in \mathrm{Set} \times \mathrm{Sub}(\mathrm{Set})$ such that $R \hookrightarrow X^2$

maps: f in Set s.t. f preserves the subobject

eg

$$\begin{tikzcd}
\mathbb{K} \arrow[r] \arrow[d] & \mathbf{Sub}(\mathbf{Set}) \arrow[d] \\
\mathbf{Set} \arrow[r, "(-)^2"'] & \mathbf{Set}
\end{tikzcd}$$

fibration
$(X, R) \mapsto X$

objects: (E, R) such that $R \hookrightarrow E$
maps: maps in Set respecting the subjects

subobject fibration
$(E, R) \mapsto E$

'change-of-base'

Abstracting away: categories of relations

idea:

1. axiomatise relations by fibrations
2. ccc-structure via structured fibrations
3. monad defined using fibration
4. restrict to concrete objects

Abstracting away: categories of relations

idea:

1. axiomatise relations by fibrations
2. ccc-structure via structured fibrations
3. monad defined using fibration
4. restrict to concrete objects

objects: $(X, R) \in \mathscr{M} \times \mathbb{E}$ such that $FX = pR$
maps: $(f, \hat{f})$ in $\mathscr{M} \times \mathbb{E}$ such that $Ff = p(\hat{f})$

fibration
$(X, R) \mapsto X$

$$\begin{tikzcd}
\mathbb{K} \arrow[r] \arrow[d, "\pi"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\
\mathcal{M} \arrow[r, "F"'] & \mathbb{B}
\end{tikzcd}$$

fibration

‘change-of-base’

Abstracting away: categories of relations

idea:

1. axiomatise relations by fibrations
2. ccc-structure via structured fibrations
3. monad defined using fibration
4. restrict to concrete objects

objects: $(X, R) \in \mathscr{M} \times \mathbb{E}$ such that $FX = pR$
maps: $(f, \hat{f})$ in $\mathscr{M} \times \mathbb{E}$ such that $Ff = p(\hat{f})$

$$
\begin{tikzcd}
\mathbb{K} \arrow[r] \arrow[d, "\pi"', "\substack{\text{fibration} \\ (X,R) \mapsto X}"' near start] \arrow[dr, phantom, "\lrcorner" very near start] & \mathbb{E} \arrow[d, "p", "\text{fibration}"'] \\
\mathcal{M} \arrow[r, "F"'] & \mathbb{B}
\end{tikzcd}
$$

'change-of-base'

Fact: if $\mathbb{E}$, $\mathbb{B}$ and $\mathscr{M}$ are CCCs, p strictly preserves CCC-structure, and F is cartesian, then:

$\mathbb{K}$ is a CCC, and π strictly preserves CCC-structure

Abstracting away: categories of relations

idea:

1. axiomatise relations by fibrations
2. ccc-structure via structured fibrations
3. monad defined using fibration
4. restrict to concrete objects

Abstracting away: categories of relations

idea:

1. axiomatise relations by fibrations
2. ccc-structure via structured fibrations
3. monad defined using fibration
4. restrict to concrete objects

eg. $\top\top$-lifting, free lifting, …

Abstracting away: categories of relations

idea:

1. axiomatise relations by fibrations
2. ccc-structure via structured fibrations
3. monad defined using fibration
4. restrict to concrete objects

Abstracting away: concreteness

(X, R_0, R_1) is concrete if every $x : 1 \to X$ in Set lifts to $(1, \top, \top) \to (X, R_0, R_1)$

Abstracting away: concreteness

(X, R_0, R_1) is concrete if every $x : 1 \to X$ in Set lifts to $(1, \top, \top) \to (X, R_0, R_1)$

$$\begin{tikzcd} \mathbb{E} \arrow[d, "p", "\text{preserves } 1"'] \\ \mathbb{B} \end{tikzcd}$$

$X \in \mathbb{E}$ is concrete if every $x : 1 \to pX$ in $\mathbb{B}$
lifts to a global element $\hat{x} : 1 \to X$ in $\mathbb{E}$

Abstracting away: concreteness

(X, R_0, R_1) is concrete if every $x : 1 \to X$ in Set lifts to $(1, \top, \top) \to (X, R_0, R_1)$

$$\begin{tikzcd} \mathbb{E} \arrow[d, "p", "\text{preserves } 1"'] \\ \mathbb{B} \end{tikzcd}$$

$X \in \mathbb{E}$ is concrete if every $x : 1 \to pX$ in $\mathbb{B}$
lifts to a global element $\hat{x} : 1 \to X$ in $\mathbb{E}$

get a subcategory $\mathrm{Conc}(\mathbb{E}) \hookrightarrow \mathbb{E}$

Categories of concrete relations (for nice enough $\mathscr{M}$)

$$\begin{tikzcd} & \mathbb{E} \arrow[d, "p"] \\ \mathcal{M} \arrow[r, "F"'] \arrow[loop below, "T"] & \mathbb{B} \end{tikzcd}$$

CCC

fibration + CCC-functor

CCC

CCC

cartesian functor

Categories of concrete relations (for nice enough $\mathscr{M}$)

$$\begin{tikzcd} \mathbb{K} \arrow[r] \arrow[d, "\pi"'] \arrow[loop above, "\hat{T}"] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\ \mathcal{M} \arrow[r, "F"'] \arrow[loop below, "T"] & \mathbb{B} \end{tikzcd}$$

monad via eg ⊤⊤-lifting

CCC

CCC

fibration + CCC-functor

CCC

CCC

cartesian functor

Categories of concrete relations (for nice enough $\mathscr{M}$)

H restricts the carrier

monad via eg $\top\top$-lifting

$$
\begin{tikzcd}
\mathrm{Conc}(\mathbb{K}) \arrow[r] & \mathbb{K} \arrow[l, bend right, "H"'] \arrow[l, bend left, "K"] \arrow[r] \arrow[d, "\pi"'] \arrow[dr, phantom, "\lrcorner", very near start] \arrow[loop above, "\hat{T}"] & \mathbb{E} \arrow[d, "p"] \\
& \mathcal{M} \arrow[r, "F"'] \arrow[loop below, "T"] & \mathbb{B}
\end{tikzcd}
$$

CCC

fibration + CCC-functor

K expands the relation

CCC

CCC

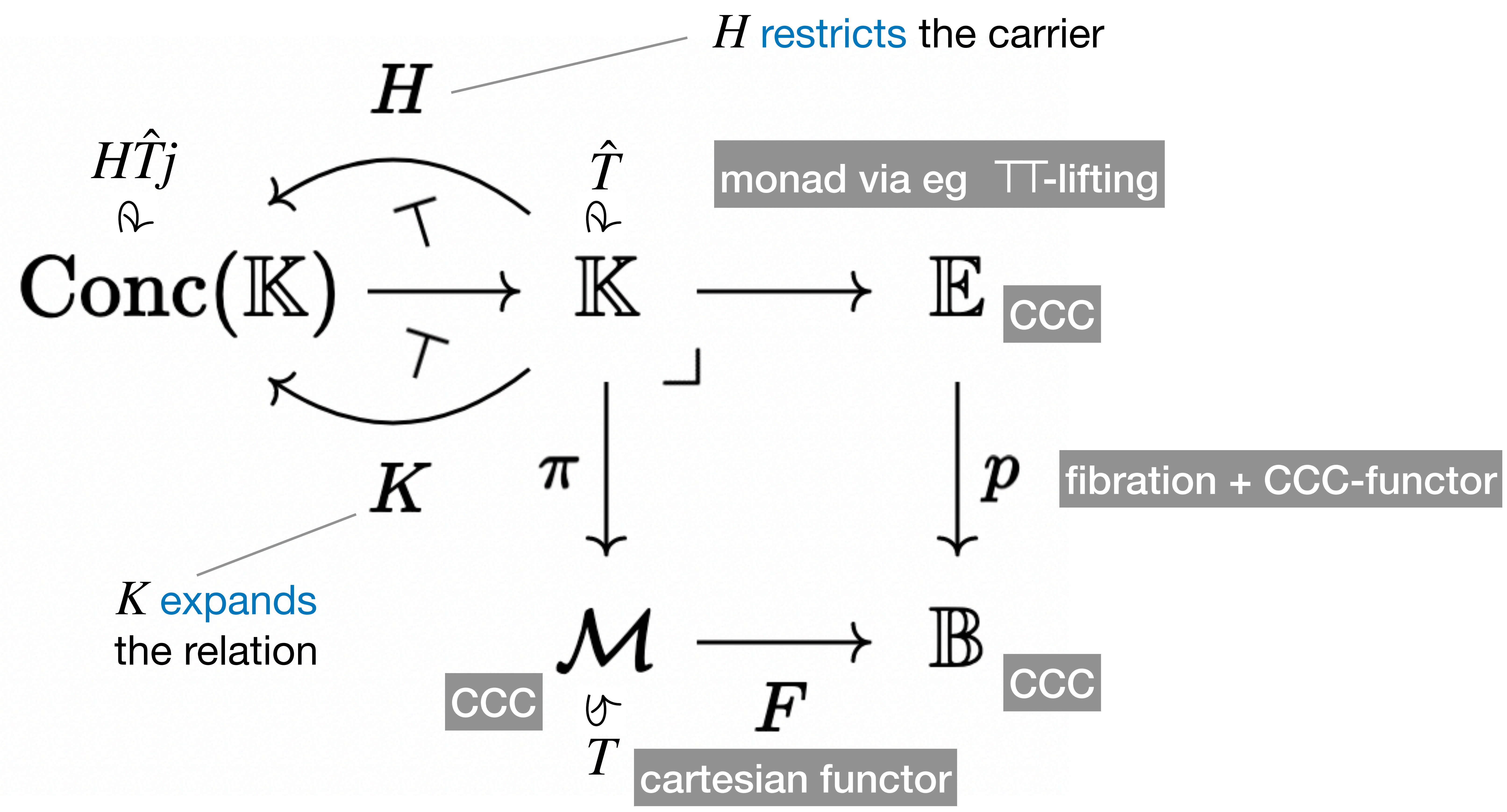
cartesian functor

Categories of concrete relations (for nice enough $\mathscr{M}$)

monad by
abstract
nonsense

ccc by
abstract
nonsense:

$[X \Rightarrow Y]_{\mathrm{Conc}} = H(jX \Rightarrow jY)$



Categories of concrete relations (for nice enough $\mathscr{M}$)

monad by abstract nonsense

ccc by abstract nonsense:

$[X \Rightarrow Y]_{\mathrm{Conc}} = H(jX \Rightarrow jY)$

$$\begin{tikzcd}
\mathrm{Conc}(\mathbb{K}) \arrow[loop above, "H\hat{T}j"] \arrow[r] & \mathbb{K} \arrow[loop above, "\hat{T}"] \arrow[l, bend right=50, "H"'] \arrow[l, bend left=50, "K"] \arrow[r] \arrow[d, "\pi"] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\
& \mathcal{M} \arrow[loop below, "T"] \arrow[r, "F"'] & \mathbb{B}
\end{tikzcd}$$

H restricts the carrier

monad via eg $\top\top$-lifting

CCC

fibration + CCC-functor

K expands the relation

CCC

CCC

cartesian functor

key property: $[X \Rightarrow Y]_{\mathrm{Conc}} \cong \mathbb{L}(jX, jY)$

internalises the preservation condition

Summing up: categories of concrete relations

Motivation

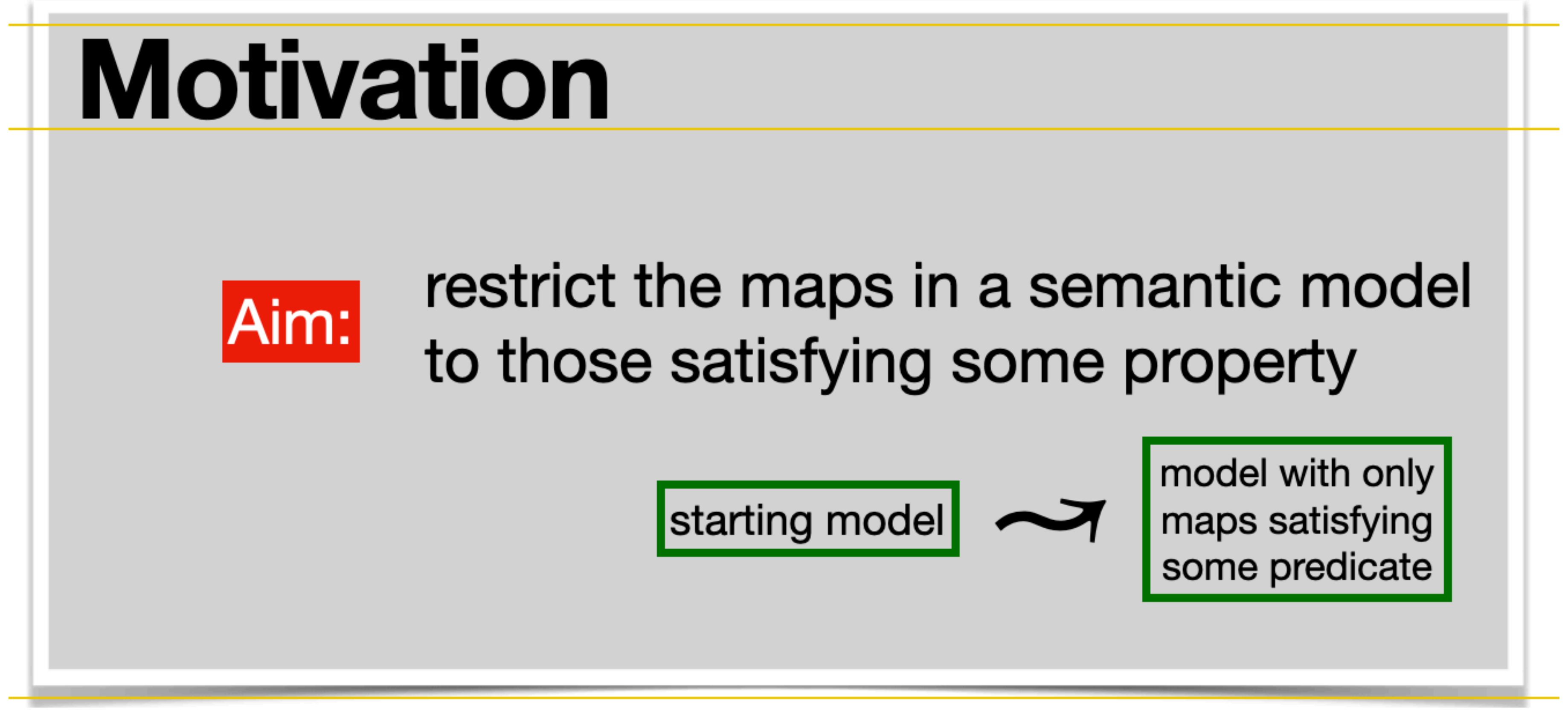
Aim: restrict the maps in a semantic model to those satisfying some property

$\boxed{\text{starting model}} \rightsquigarrow \boxed{\text{model with only maps satisfying some predicate}}$

Summing up: categories of concrete relations

idea:

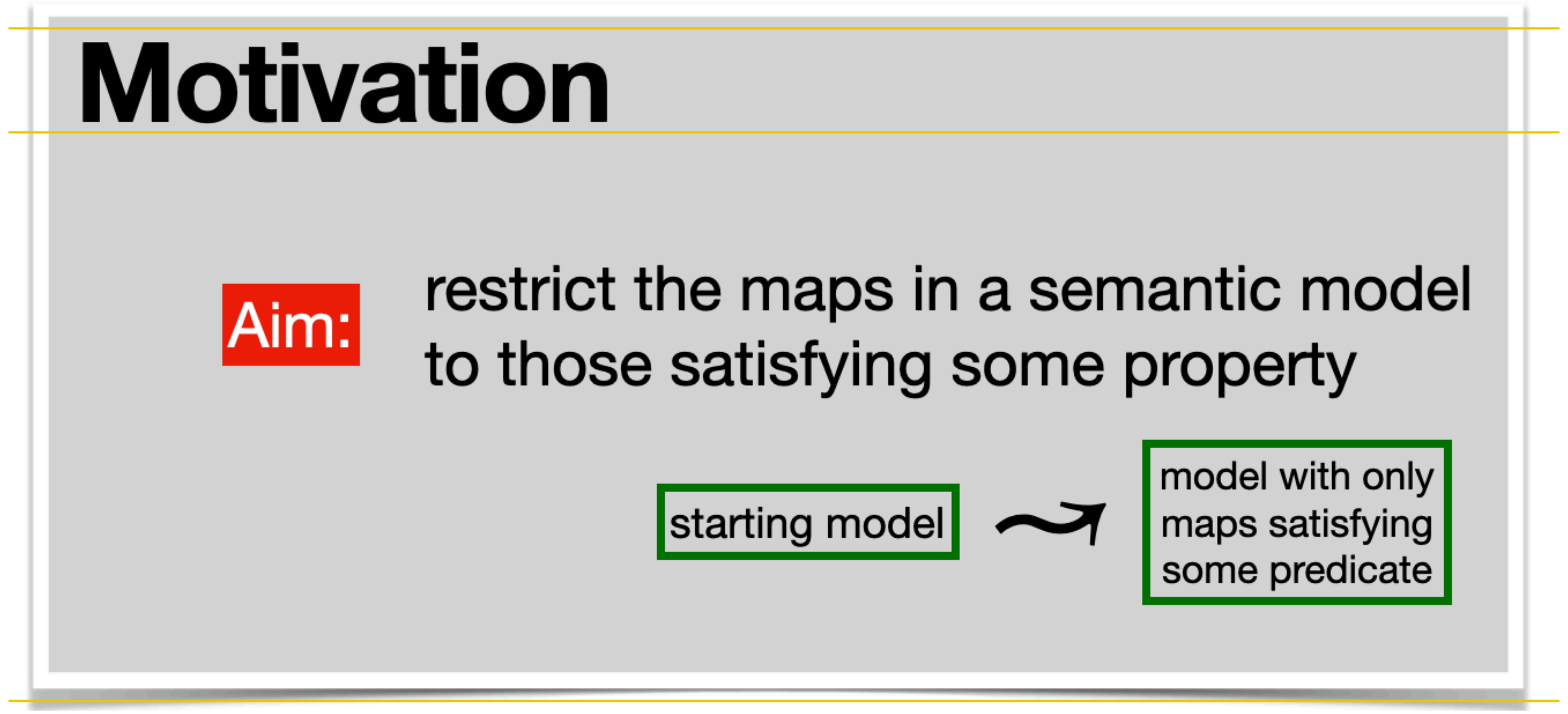
1. axiomatise relations by fibrations
2. ccc-structure via structured fibrations
3. monad defined using fibration
4. restrict to concrete objects



Summing up: categories of concrete relations

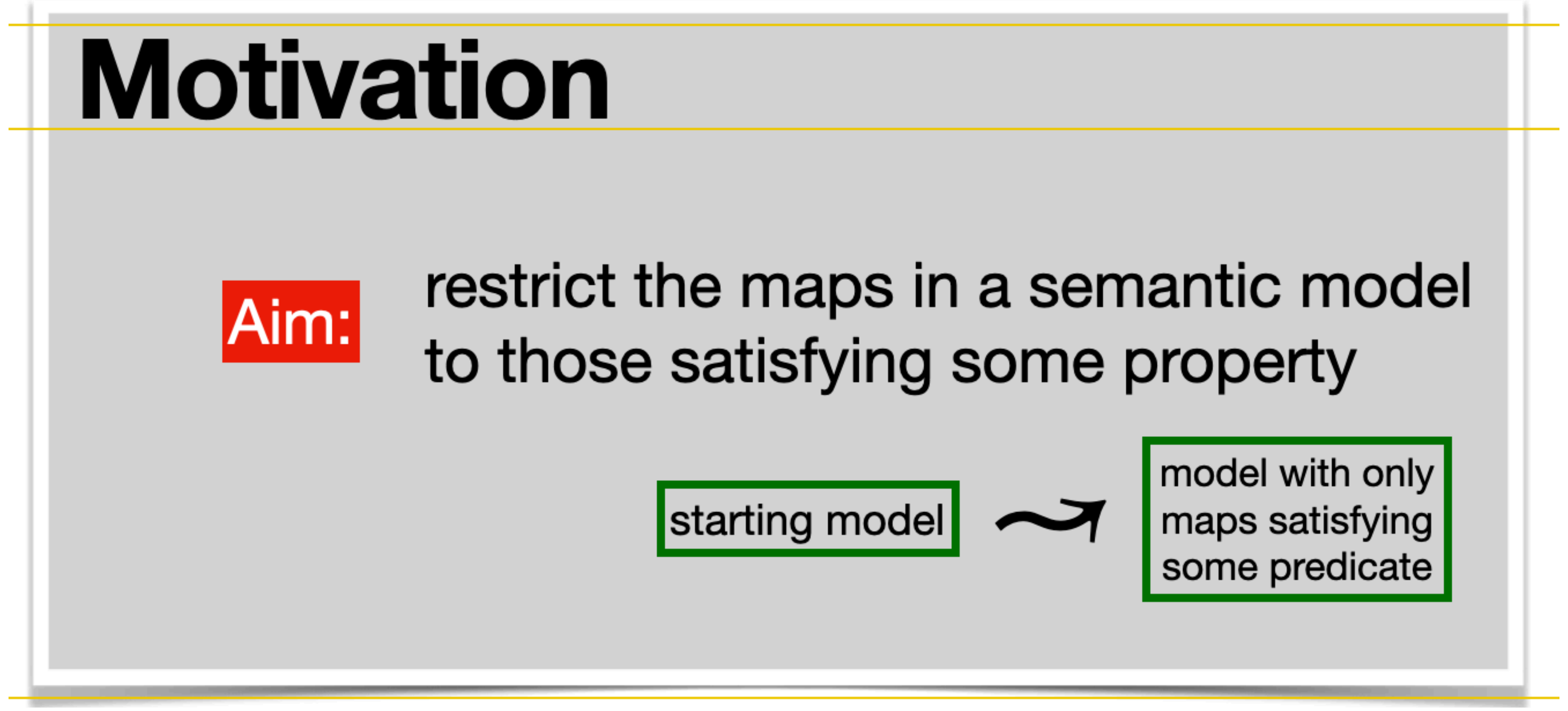
idea:

1. axiomatise relations by fibrations
2. ccc-structure via structured fibrations
3. monad defined using fibration
4. restrict to concrete objects



the induced model $\left(\mathrm{Conc}(\mathbb{K}), H\hat{T}j, \hat{s}\right)$ **restricts to maps preserving the 'relations' encoded by** p

Summing up: categories of concrete relations



idea:

1. axiomatise relations by fibrations
2. ccc-structure via structured fibrations
3. monad defined using fibration
4. restrict to concrete objects

the induced model $\big(\mathrm{Conc}(\mathbb{K}), H\hat{T}j, \hat{s}\big)$
restricts to maps preserving the 'relations'
encoded by p

in fact, encodes
preservation of a
logical relation

2: Logical relations

What is a logical relation?

the classical story:
Plotkin + many others

logical relation R =

A family of relations $\{R_\sigma \mid \sigma \in \mathrm{Type}\}$ such that:

(1) R_σ is a relation on $[\![\sigma]\!]$

(2) the family is compatible with the language's type structure

What is a logical relation?

the classical story:
Plotkin + many others

logical relation R =

A family of relations $\{R_\sigma \mid \sigma \in \mathrm{Type}\}$ such that:

(1) R_σ is a relation on $[\![\sigma]\!]$

(2) the family is compatible with the language's type structure

Basic Lemma =

$$(M : \sigma) \implies [\![M]\!] \in R_\sigma$$

useful for relating models, or proving facts about models

What is a logical relation?

the classical story:
Plotkin + many others

logical relation R =

A family of relations $\{R_\sigma \mid \sigma \in \mathrm{Type}\}$
such that:

(1) R_σ is a relation on $[\![\sigma]\!]$

(2) the family is compatible with the language's type structure

Basic Lemma* =

f is definable $\iff$ f 'satisfies' every logical relation

useful for relating models, or proving facts about models

Logical relations for simply-typed lambda calculus

(the classical story)

$R_\sigma \subseteq [\![\sigma]\!]^n$ for each type σ, and

Logical relations for simply-typed lambda calculus

(the classical story)

$R_\sigma \subseteq [\![\sigma]\!]^n$ for each type σ, and

$$(f_1, \ldots, f_n) \in (R \supset S) \iff \big((x_1, \ldots, x_n) \in R \implies (f_1\, x_1, \ldots, f_n\, x_n) \in S\big)$$

- exponentials: $R_{\sigma\to\tau} = (R_\sigma \supset R_\tau)$

Logical relations for simply-typed lambda calculus

(the classical story)

$R_\sigma \subseteq [\![\sigma]\!]^n$ for each type σ, and

$$(f_1, \ldots, f_n) \in (R \supset S) \iff \big((x_1, \ldots, x_n) \in R \implies (f_1\, x_1, \ldots, f_n\, x_n) \in S\big)$$

- exponentials: $R_{\sigma \to \tau} = (R_\sigma \supset R_\tau)$
- terminal object: $R_1 = \top$

$$\top = \{(\bullet, \ldots, \bullet)\}$$

Logical relations for simply-typed lambda calculus

(the classical story)

$R_\sigma \subseteq [\![\sigma]\!]^n$ for each type σ, and

$(f_1, \ldots, f_n) \in (R \supset S)$
$\iff \big((x_1, \ldots, x_n) \in R \implies (f_1\, x_1, \ldots, f_n\, x_n) \in S\big)$

- exponentials: $R_{\sigma\to\tau} = (R_\sigma \supset R_\tau)$
- terminal object: $R_1 = \top$
- products: $R_{\sigma_1\times\sigma_2} = R_{\sigma_1} \star R_{\sigma_2}$

$\top = \{(\bullet, \ldots, \bullet)\}$

$\big((x_1, y_1), \ldots, (x_n, y_n)\big) \in (R \star S)$
$\iff (x_1, \ldots, x_n) \in R$ and $(y_1, \ldots, y_n) \in S$

Logical relations for simply-typed lambda calculus

(the classical story)

$R_\sigma \subseteq [\![\sigma]\!]^n$ for each type σ, and

$(f_1, \ldots, f_n) \in (R \supset S)$
$\iff \big((x_1, \ldots, x_n) \in R \implies (f_1\, x_1, \ldots, f_n\, x_n) \in S\big)$

- exponentials: $R_{\sigma \to \tau} = (R_\sigma \supset R_\tau)$
- terminal object: $R_1 = \top$
- products: $R_{\sigma_1 \times \sigma_2} = R_{\sigma_1} \star R_{\sigma_2}$

$\top = \{(\bullet, \ldots, \bullet)\}$

$\big((x_1, y_1), \ldots, (x_n, y_n)\big) \in (R \star S)$
$\iff (x_1, \ldots, x_n) \in R$ and $(y_1, \ldots, y_n) \in S$

what's a principled extension to monadic structure?

What is a logical relation? (Hermida, Jacobs, ...)

logical relation

$$(\text{free model}) \xrightarrow[\boldsymbol{s}[\![-]\!]]{} \mathcal{M}$$

interpretation
induced by
$s : \mathrm{Base} \to \mathscr{M}$

semantic model
= CCC

What is a logical relation? (Hermida, Jacobs, ...)

logical relation

$$\begin{tikzcd} & \mathbb{E} \arrow[d, "p"] \\ \text{(free model)} \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \end{tikzcd}$$

CCC

fibration &
strict CCC functor

semantic model
= CCC

interpretation
induced by
$s : \mathrm{Base} \to \mathscr{M}$

What is a logical relation? (Hermida, Jacobs, ...)

logical relation

semantic interpretation

CCC

fibration &
strict CCC functor

$$
\begin{tikzcd}
 & \mathbb{E} \arrow[d, "p"] \\
\text{(free model)} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M}
\end{tikzcd}
$$

interpretation
induced by
$s : \mathrm{Base} \to \mathscr{M}$

semantic model
= CCC

What is a logical relation? (Hermida, Jacobs, ...)

logical relation

semantic interpretation

CCC

$$
\begin{tikzcd}
 & \mathbb{E} \arrow[d, "p"] \\
\text{(free model)} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M}
\end{tikzcd}
$$

fibration &
strict CCC functor

interpretation
induced by
$s : \mathrm{Base} \to \mathscr{M}$

semantic model
= CCC

$p(\hat{s}[\![\sigma]\!]) = s[\![\sigma]\!]$ for all types σ

What is a logical relation? The canonical example

(Hermida, Jacobs, …)

$$\text{(free model)} \xrightarrow[\boldsymbol{s}[\![-]\!]]{} \mathcal{M}$$

interpretation
induced by
$s : \mathrm{Base} \to \mathscr{M}$

What is a logical relation? The canonical example

(Hermida, Jacobs, …)

CCC

$$\begin{tikzcd}
 & & \mathbb{E} \arrow[d, "p"] \\
\text{(free model)} \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \arrow[r, "F"'] & \mathbb{B}
\end{tikzcd}$$

fibration &
strict CCC functor

CCC

interpretation
induced by
$s : \mathrm{Base} \to \mathscr{M}$

cartesian functor

What is a logical relation? The canonical example

(Hermida, Jacobs, …)

CCC objects: pairs (X, R)
maps: maps preserving the relation

CCC

$$
\begin{tikzcd}
 & \mathbb{K} \arrow[r] \arrow[d, "\pi"] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\
\text{(free model)} \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \arrow[r, "F"'] & \mathbb{B}
\end{tikzcd}
$$

fibration &
strict CCC
functor

fibration &
strict CCC functor

CCC

cartesian functor

interpretation
induced by
$s : \text{Base} \to \mathscr{M}$

What is a logical relation? The canonical example

(Hermida, Jacobs, …)

semantic interpretation

$\hat{s}[\![\beta]\!] := (s[\![\beta]\!], R_\beta)$

CCC

objects: pairs (X, R)

maps: maps preserving the relation

CCC

$$
\begin{tikzcd}
 & \mathbb{K} \arrow[r] \arrow[d, "\pi"] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\
\text{(free model)} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \arrow[r, "F"'] & \mathbb{B}
\end{tikzcd}
$$

fibration & strict CCC functor

CCC

interpretation induced by $s : \mathrm{Base} \to \mathscr{M}$

cartesian functor

What is a logical relation? The canonical example

(Hermida, Jacobs, …)

semantic interpretation

$\hat{s}[\![\beta]\!] := (s[\![\beta]\!], R_\beta)$

CCC objects: pairs (X, R)
maps: maps preserving the relation

$$\begin{tikzcd} & \mathbb{K} \arrow[r] \arrow[d, "\pi"] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\ (\text{free model}) \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \arrow[r, "F"'] & \mathbb{B} \end{tikzcd}$$

CCC

fibration &
strict CCC functor

CCC

interpretation
induced by
$s : \mathrm{Base} \to \mathscr{M}$

cartesian functor

$$\hat{s}[\![\sigma \to \tau]\!] = \big(s[\![\sigma]\!] \Rightarrow s[\![\tau]\!], R_\sigma \supset R_\tau\big)$$

$$\hat{s}[\![\sigma_1 \times \sigma_2]\!] = \big(s[\![\sigma_1]\!] \times s[\![\sigma_2]\!], R_{\sigma_1} \star R_{\sigma_2}\big)$$

Logical relations for simply-typed lambda calculus

Some examples:

- $\mathrm{SN}_\sigma = \{[\![M]\!] \mid M : \sigma$ is strongly normalising$\}$
- $\mathrm{Eq}_\sigma(\Gamma) = \left\{([\![M]\!], [\![M']\!]) \mid \Gamma \vdash M \simeq M' : \sigma\right\}$

for a 'suitable' equational theory

- $\mathrm{Def}_\sigma(\Gamma) = \{[\![M]\!] \mid \Gamma \vdash M : \sigma\}$

note the parametrisation by contexts

Kripke relations of varying arity

(for a CCC $\mathscr{M}$)

$\mathrm{Def}_\sigma(\Gamma) = \{ [\![M]\!] \mid \Gamma \vdash M : \sigma \} \subseteq \mathscr{M}([\![\Gamma]\!], [\![\sigma]\!])$

satisfies monotonicity:

$$[\![M]\!] \in \mathrm{Def}_\sigma(\Gamma) \text{ and } \Gamma \subseteq \Delta \implies [\![M^{\mathrm{wkn}}]\!] \in \mathrm{Def}_\sigma(\Delta)$$

Def_σ is a presheaf over a category of contexts

Kripke relations of varying arity

(for a CCC $\mathscr{M}$)

$\mathrm{Def}_\sigma(\Gamma) = \{[\![M]\!] \mid \Gamma \vdash M : \sigma\} \subseteq \mathscr{M}([\![\Gamma]\!], [\![\sigma]\!])$

satisfies monotonicity:

$[\![M]\!] \in \mathrm{Def}_\sigma(\Gamma)$ and $\Gamma \subseteq \Delta \implies [\![M^{\mathrm{wkn}}]\!] \in \mathrm{Def}_\sigma(\Delta)$

Def_σ is a sub-presheaf of $\mathscr{M}([\![-]\!], [\![\sigma]\!])$

Kripke relations of varying arity

(for a CCC $\mathscr{M}$)

$\mathrm{Def}_\sigma(\Gamma) = \{[\![M]\!] \mid \Gamma \vdash M : \sigma\} \subseteq \mathscr{M}([\![\Gamma]\!], [\![\sigma]\!])$

satisfies monotonicity:

$[\![M]\!] \in \mathrm{Def}_\sigma(\Gamma)$ and $\Gamma \subseteq \Delta \implies [\![M^{\mathrm{wkn}}]\!] \in \mathrm{Def}_\sigma(\Delta)$

Def_σ is a sub-presheaf of $\mathscr{M}([\![-]\!], [\![\sigma]\!])$

A Kripke relation of varying arity on $X \in \mathscr{M}$

is a subpresheaf $R \hookrightarrow \mathscr{M}([\![-]\!], X)$

$R(\Gamma) \subseteq \mathscr{M}([\![\Gamma]\!], X)$

$f \in R(\Gamma)$ and $\Gamma \subseteq \Delta \implies f \circ [\![\mathrm{wkn}]\!] \in R(\Delta)$

Kripke relations of varying arity

(for a CCC $\mathscr{M}$)

$\mathrm{Def}_\sigma(\Gamma) = \{ [\![M]\!] \mid \Gamma \vdash M : \sigma \} \subseteq \mathscr{M}([\![\Gamma]\!], [\![\sigma]\!])$

satisfies monotonicity:

$[\![M]\!] \in \mathrm{Def}_\sigma(\Gamma)$ and $\Gamma \subseteq \Delta \implies [\![M^{\mathrm{wkn}}]\!] \in \mathrm{Def}_\sigma(\Delta)$

Def_σ is a sub-presheaf of $\mathscr{M}([\![-]\!], [\![\sigma]\!])$

A Kripke relation of varying arity on $X \in \mathscr{M}$

is a subpresheaf $R \hookrightarrow \mathscr{M}([\![-]\!], X)$

$R(\Gamma) \subseteq \mathscr{M}([\![\Gamma]\!], X)$

$f \in R(\Gamma)$ and $\Gamma \subseteq \Delta \implies f \circ [\![\mathrm{wkn}]\!] \in R(\Delta)$

$$\begin{tikzcd}
\mathrm{Krip} \arrow[r] \arrow[d, "\pi"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Sub}(\widehat{\mathrm{Con}}) \arrow[d, "\mathrm{cod}"] \\
\mathcal{M} \arrow[r, "X \mapsto \mathcal{M}([\![-]\!], X)"'] & \widehat{\mathrm{Con}}
\end{tikzcd}$$

What is a logical relation? (Hermida, Jacobs, ...)

logical relation

semantic interpretation

CCC

$$\begin{tikzcd} & \mathbb{E} \arrow[d, "p"] \\ \text{(free model)} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \end{tikzcd}$$

fibration & strict CCC functor

semantic model = CCC

interpretation induced by $s : \mathrm{Base} \to \mathscr{M}$

Basic Lemma

$$(M : \sigma) \implies [\![M]\!] \in R_\sigma$$

What is a logical relation? (Hermida, Jacobs, ...)

logical relation

semantic interpretation

CCC

fibration &
strict CCC functor

$$\begin{tikzcd} & \mathbb{E} \arrow[d, "p"] \\ \text{(free model)} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \end{tikzcd}$$

semantic model
= CCC

interpretation
induced by
$s : \mathrm{Base} \to \mathscr{M}$

Basic Lemma

$f : s[\![\Gamma]\!] \to s[\![\sigma]\!]$ satisfies R $\iff$ f lifts to a map in $\mathbb{E}$

($f = p(\hat{f})$ for some $\hat{f}$)

What is a logical relation? (Hermida, Jacobs, ...)

logical relation

semantic interpretation

CCC

$$
\begin{tikzcd}
 & \mathbb{E} \arrow[d, "p"] \\
\text{(free model)} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M}
\end{tikzcd}
$$

fibration & strict CCC functor

semantic model = CCC

interpretation induced by $s : \mathrm{Base} \to \mathscr{M}$

Basic Lemma

$f : s[\![\Gamma]\!] \to s[\![\sigma]\!]$ satisfies R $\iff$ f lifts to a map in $\mathbb{E}$

($f = p(\hat{f})$ for some $\hat{f}$)

f is definable $\implies$ f satisfies R

Proof: $f = s[\![M]\!] \implies f = p(\hat{s}[\![M]\!])$

What is a logical relation? (a 2-categorical perspective — WIP)

logical relation

semantic interpretation

$$\begin{tikzcd} & \mathbb{E} \arrow[d, "p"] \\ \text{(free model)} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \end{tikzcd}$$

internal fibration in 2-category of CCCs and strict CC-functors

semantic model

interpretation induced by $s : \mathrm{Base} \to \mathscr{M}$

What is a logical relation? (a 2-categorical perspective — WIP)

logical relation

semantic interpretation

$$
\begin{tikzcd}
 & \mathbb{E} \arrow[d, "p"] \\
\text{(free model)} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M}
\end{tikzcd}
$$

internal fibration in 2-category of models and semantics-preserving functors

semantic model

interpretation induced by $s : \mathrm{Base} \to \mathscr{M}$

What is a logical relation? (a 2-categorical perspective — WIP)

logical relation

semantic interpretation

$$\begin{tikzcd} & \hat{T} \arrow[d, phantom, "\circlearrowright"] \\ & \mathbb{E} \arrow[d, "p"] \\ (\textbf{free model}) \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \\ & T \arrow[u, phantom, "\circlearrowleft"] \end{tikzcd}$$

internal fibration in 2-category of models and semantics-preserving functors

semantic model

interpretation induced by $s : \mathrm{Base} \to \mathscr{M}$

Basic Lemma

$f : s[\![\Gamma]\!] \to s[\![\sigma]\!]$ satisfies R $\iff$ f lifts to a map in $\mathbb{E}$

($f = p(\hat{f})$ for some $\hat{f}$)

f is definable $\implies$ f satisfies R

What is a logical relation for λ_{ml}?

logical relation

semantic interpretation

$$
\begin{tikzcd}
 & \hat{T} \arrow[d, phantom, "\circlearrowleft"] \\
 & \mathbb{E} \arrow[d, "p"] \\
\text{(free model)} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \arrow[d, phantom, "\circlearrowleft"] \\
 & T
\end{tikzcd}
$$

internal fibration in 2-category of models and semantics-preserving functors

semantic model

interpretation induced by $s : \mathrm{Base} \to \mathscr{M}$

What is a logical relation for λ_{ml}?

logical relation

semantic interpretation

$$
\begin{tikzcd}
 & \mathbb{E} \arrow[loop above, "\hat{T}"] \arrow[d, "p"] \\
(\text{free model}) \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \arrow[loop below, "T"]
\end{tikzcd}
$$

internal fibration in 2-category of models and semantics-preserving functors

semantic model

interpretation induced by $s : \mathrm{Base} \to \mathscr{M}$

logical relation

Fibration p such that

- p strictly preserves cc-structure
- p commutes with the monads: $p \circ \hat{T} = T \circ p, \ldots$

What is a logical relation for λ_{ml}?

semantic interpretation

$\hat{s}[\![\beta]\!] := (s[\![\beta]\!], R_\beta)$

CCC objects: pairs (X, R)
maps: maps preserving the relation

$$
\begin{tikzcd}
 & \mathbb{K} \arrow[r] \arrow[d, "\pi"] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\
(\text{free model}) \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \arrow[r, "F"'] \arrow[loop below, "T"] & \mathbb{B}
\end{tikzcd}
$$

CCC (for $\mathbb{E}$)

fibration &
strict CCC functor

CCC (for $\mathbb{B}$)

cartesian functor

interpretation
induced by
$s : \text{Base} \to \mathscr{M}$

$$\hat{s}[\![\sigma \to \tau]\!] = \big(s[\![\sigma]\!] \Rightarrow s[\![\tau]\!], R_\sigma \supset R_\tau\big)$$

$$\hat{s}[\![\sigma_1 \times \sigma_2]\!] = \big(s[\![\sigma_1]\!] \times s[\![\sigma_2]\!], R_{\sigma_1} \star R_{\sigma_2}\big)$$

What is a logical relation for λ_{ml}?

monad defined using fibration eg. ⊤⊤-lifting, free lifting, …

CCC objects: pairs (X, R)
maps: maps preserving the relation

semantic interpretation

$\hat{s}[\![\beta]\!] := (s[\![\beta]\!], R_\beta)$

$$\begin{tikzcd}
& \hat{T} \circlearrowright \mathbb{K} \arrow[r] \arrow[d, "\pi"] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\
(\text{free model}) \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \circlearrowleft T \arrow[r, "F"'] & \mathbb{B}
\end{tikzcd}$$

CCC (for $\mathbb{E}$)

fibration &
strict CCC functor

CCC (for $\mathbb{B}$)

interpretation
induced by
$s : \mathrm{Base} \to \mathscr{M}$

cartesian functor

$$\hat{s}[\![\sigma \to \tau]\!] = \left(s[\![\sigma]\!] \Rightarrow s[\![\tau]\!], R_\sigma \supset R_\tau\right)$$

$$\hat{s}[\![\sigma_1 \times \sigma_2]\!] = \left(s[\![\sigma_1]\!] \times s[\![\sigma_2]\!], R_{\sigma_1} \star R_{\sigma_2}\right)$$

What is a logical relation for λ_{ml}?

monad defined using fibration eg. ⊤⊤ -lifting, free lifting, …

CCC objects: pairs (X, R)
maps: maps preserving the relation

semantic interpretation

$$\hat{s}[\![\beta]\!] := (s[\![\beta]\!], R_\beta)$$

CCC

fibration &
strict CCC functor

$$\begin{tikzcd}
& \mathbb{K} \arrow["\hat{T}"', loop left] \arrow[r] \arrow[d, "\pi"] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\
\text{(free model)} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathcal{M} \arrow["T"', loop below] \arrow[r, "F"'] & \mathbb{B}
\end{tikzcd}$$

interpretation
induced by
$s : \mathrm{Base} \to \mathscr{M}$

CCC

cartesian functor

$$\hat{s}[\![\sigma \to \tau]\!] = \big(s[\![\sigma]\!] \Rightarrow s[\![\tau]\!], R_\sigma \supset R_\tau\big)$$

$$\hat{s}[\![\sigma_1 \times \sigma_2]\!] = \big(s[\![\sigma_1]\!] \times s[\![\sigma_2]\!], R_{\sigma_1} \star R_{\sigma_2}\big)$$

$$\boxed{\hat{s}[\![T\sigma]\!] = \big(Ts[\![\sigma]\!], \hat{T}R_\sigma\big)}$$

$$\hat{s}[\![\sigma \to \tau]\!] = \left(s[\![\sigma]\!] \Rightarrow s[\![\tau]\!], R_\sigma \supset R_\tau\right)$$

$$\hat{s}[\![\sigma_1 \times \sigma_2]\!] = \left(s[\![\sigma_1]\!] \times s[\![\sigma_2]\!], R_{\sigma_1} \star R_{\sigma_2}\right)$$

$$\boxed{\hat{s}[\![T\sigma]\!] = \left(Ts[\![\sigma]\!], \hat{T}R_\sigma\right)}$$

<u>Some examples:</u>

- $\mathrm{SN}_\sigma = \{[\![M]\!] \mid M : \sigma$ is strongly normalising$\}$
- $\mathrm{Eq}_\sigma(\Gamma) = \left\{([\![M]\!], [\![M']\!]) \mid \Gamma \vdash M \simeq M' : \sigma\right\}$

for a 'suitable' equational theory

- $\mathrm{Def}_\sigma(\Gamma) = \{[\![M]\!] \mid \Gamma \vdash M : \sigma\}$

note the parametrisation by contexts

difficulty = choice of monad $\hat{T}$

⊤⊤-lifting very useful for this (cf. Lindley-Stark, biorthogonality,…)

What is a logical relation for λ_c?

$$\boxed{\begin{aligned} s[\![\sigma \to \tau]\!] &= s[\![\sigma]\!] \Rightarrow T(s[\![\tau]\!]) \\ s[\![\sigma_1 \times \sigma_2]\!] &= s[\![\sigma_1]\!] \times s[\![\sigma_2]\!] \end{aligned}}$$

What is a logical relation for λ_c?

$$\hat{s}[\![\sigma \to \tau]\!] = \left(s[\![\sigma]\!] \Rightarrow s[\![\tau]\!], R_\sigma \supset R_\tau\right)$$

$$\hat{s}[\![\sigma_1 \times \sigma_2]\!] = \left(s[\![\sigma_1]\!] \times s[\![\sigma_2]\!], R_{\sigma_1} \star R_{\sigma_2}\right)$$

$$s[\![\sigma \to \tau]\!] = s[\![\sigma]\!] \Rightarrow T(s[\![\tau]\!])$$

$$s[\![\sigma_1 \times \sigma_2]\!] = s[\![\sigma_1]\!] \times s[\![\sigma_2]\!]$$

What is a logical relation for λ_c?

$$\cancel{\begin{aligned}\hat{s}[\![\sigma \to \tau]\!] &= \left(s[\![\sigma]\!] \Rightarrow s[\![\tau]\!], R_\sigma \supset R_\tau\right) \\ \hat{s}[\![\sigma_1 \times \sigma_2]\!] &= \left(s[\![\sigma_1]\!] \times s[\![\sigma_2]\!], R_{\sigma_1} \star R_{\sigma_2}\right)\end{aligned}}$$

$$\begin{aligned}s[\![\sigma \to \tau]\!] &= s[\![\sigma]\!] \Rightarrow T(s[\![\tau]\!]) \\ s[\![\sigma_1 \times \sigma_2]\!] &= s[\![\sigma_1]\!] \times s[\![\sigma_2]\!]\end{aligned}$$

Restrict to values:

for $R \subseteq (T[\![\sigma]\!])^n$,

$$R^{\text{vals}} := \{(x_1, \ldots, x_n) \mid (\eta\, x_1, \ldots, \eta\, x_n) \in R\} \subseteq [\![\sigma]\!]^n$$

$$\boxed{\begin{aligned}\hat{s}[\![\sigma \to \tau]\!] &= \left(s[\![\sigma]\!] \Rightarrow Ts[\![\tau]\!], R_\sigma^{\text{vals}} \supset R_\tau\right) \\ \hat{s}[\![\sigma_1 \times \sigma_2]\!] &= \left(s[\![\sigma_1]\!] \times s[\![\sigma_2]\!], R_{\sigma_1}^{\text{vals}} \star R_{\sigma_2}^{\text{vals}}\right)\end{aligned}}$$

What is a logical relation for λ_c?

$$\cancel{\boxed{\begin{aligned}\hat{s}[\![\sigma \to \tau]\!] &= \left(s[\![\sigma]\!] \Rightarrow s[\![\tau]\!], R_\sigma \supset R_\tau\right) \\ \hat{s}[\![\sigma_1 \times \sigma_2]\!] &= \left(s[\![\sigma_1]\!] \times s[\![\sigma_2]\!], R_{\sigma_1} \star R_{\sigma_2}\right)\end{aligned}}}$$

$$\boxed{\begin{aligned}s[\![\sigma \to \tau]\!] &= s[\![\sigma]\!] \Rightarrow T(s[\![\tau]\!]) \\ s[\![\sigma_1 \times \sigma_2]\!] &= s[\![\sigma_1]\!] \times s[\![\sigma_2]\!]\end{aligned}}$$

Restrict to values:

for $R \subseteq (T[\![\sigma]\!])^n$,

$$R^{\text{vals}} := \{(x_1, \ldots, x_n) \mid (\eta\, x_1, \ldots, \eta\, x_n) \in R\} \subseteq [\![\sigma]\!]^n$$

$$\boxed{\begin{aligned}\hat{s}[\![\sigma \to \tau]\!] &= \left(s[\![\sigma]\!] \Rightarrow Ts[\![\tau]\!], R_\sigma^{\text{vals}} \supset R_\tau\right) \\ \hat{s}[\![\sigma_1 \times \sigma_2]\!] &= \left(s[\![\sigma_1]\!] \times s[\![\sigma_2]\!], R_{\sigma_1}^{\text{vals}} \star R_{\sigma_2}^{\text{vals}}\right)\end{aligned}}$$

$$\boxed{\hat{s}[\![\sigma]\!] = \left(s[\![\sigma]\!], R_\sigma^{\text{vals}}\right)}$$

What is a logical relation for λ_c?

$$\cancel{\hat{s}[\![\sigma \to \tau]\!] = \left(s[\![\sigma]\!] \Rightarrow s[\![\tau]\!], R_\sigma \supset R_\tau\right)}$$

$$\cancel{\hat{s}[\![\sigma_1 \times \sigma_2]\!] = \left(s[\![\sigma_1]\!] \times s[\![\sigma_2]\!], R_{\sigma_1} \star R_{\sigma_2}\right)}$$

$$s[\![\sigma \to \tau]\!] = s[\![\sigma]\!] \Rightarrow T(s[\![\tau]\!])$$

$$s[\![\sigma_1 \times \sigma_2]\!] = s[\![\sigma_1]\!] \times s[\![\sigma_2]\!]$$

Restrict to values:

for $R \subseteq (T[\![\sigma]\!])^n$,

$$R^{\text{vals}} := \{(x_1, \dots, x_n) \mid (\eta\, x_1, \dots, \eta\, x_n) \in R\} \subseteq [\![\sigma]\!]^n$$

$$\boxed{\begin{aligned}\hat{s}[\![\sigma \to \tau]\!] &= \left(s[\![\sigma]\!] \Rightarrow Ts[\![\tau]\!], R_\sigma^{\text{vals}} \supset R_\tau\right)\\ \hat{s}[\![\sigma_1 \times \sigma_2]\!] &= \left(s[\![\sigma_1]\!] \times s[\![\sigma_2]\!], R_{\sigma_1}^{\text{vals}} \star R_{\sigma_2}^{\text{vals}}\right)\end{aligned}}$$

$$\boxed{\begin{aligned}\hat{s}[\![\sigma]\!] &= \left(s[\![\sigma]\!], R_\sigma^{\text{vals}}\right)\\ T\hat{s}[\![\sigma]\!] &= \left(Ts[\![\sigma]\!], R_\sigma\right)\end{aligned}}$$

$$\boxed{\hat{T}\left(s[\![\sigma]\!], R_\sigma^{\text{vals}}\right) = \left(Ts[\![\sigma]\!], R_\sigma\right)}$$

Converse to the Basic Lemma

Every morphism of models defines a logical relation:

$$
\begin{tikzcd}
 & (\text{free model}) \arrow[dl, "c[\![-]\!]"'] \arrow[dd, "s[\![-]\!]"] \\
S \circlearrowleft \mathcal{C} \arrow[dr, "F"'] & \\
 & \mathcal{M} \\
 & \circlearrowleft \\
 & T
\end{tikzcd}
$$

Converse to the Basic Lemma

Every morphism of models defines a logical relation:

$$\mathscr{F}_\sigma(\Gamma) := \{Fh \mid h \in \mathscr{C}(c[\![\Gamma]\!], c[\![\sigma]\!])\}$$

$$\begin{tikzcd}
 & \text{(free model)} \arrow[dl, "{\boldsymbol{c}[\![-]\!]}"'] \arrow[dd, "{\boldsymbol{s}[\![-]\!]}"] \\
S \circlearrowright \mathcal{C} \arrow[dr, "\boldsymbol{F}"'] & \\
 & \mathcal{M} \\
 & T \arrow[u, phantom, "\circlearrowleft" description]
\end{tikzcd}$$

Converse to the Basic Lemma

Every morphism of models defines a logical relation:

$$\mathscr{F}_\sigma(\Gamma) := \{Fh \mid h \in \mathscr{C}(c[\![\Gamma]\!], c[\![\sigma]\!])\}$$

$$\begin{tikzcd}
 & \text{(free model)} \arrow[dl, "{c[\![-]\!]}"'] \arrow[dr, "{\sigma \mapsto (s[\![\sigma]\!], \mathscr{F}_\sigma)}"] & \\
S \circlearrowright \mathcal{C} \arrow[rr, "Y", "{C \mapsto (FC, \mathscr{F}_C)}"'] \arrow[dr, "F"'] & & \mathbf{Krip} \circlearrowleft \hat{T} \arrow[dl, "\pi"] \\
 & \mathcal{M} \circlearrowleft T &
\end{tikzcd}$$

monad given by ⊤⊤ -lifting

cf. the Identity Extension Lemma

Converse to the Basic Lemma

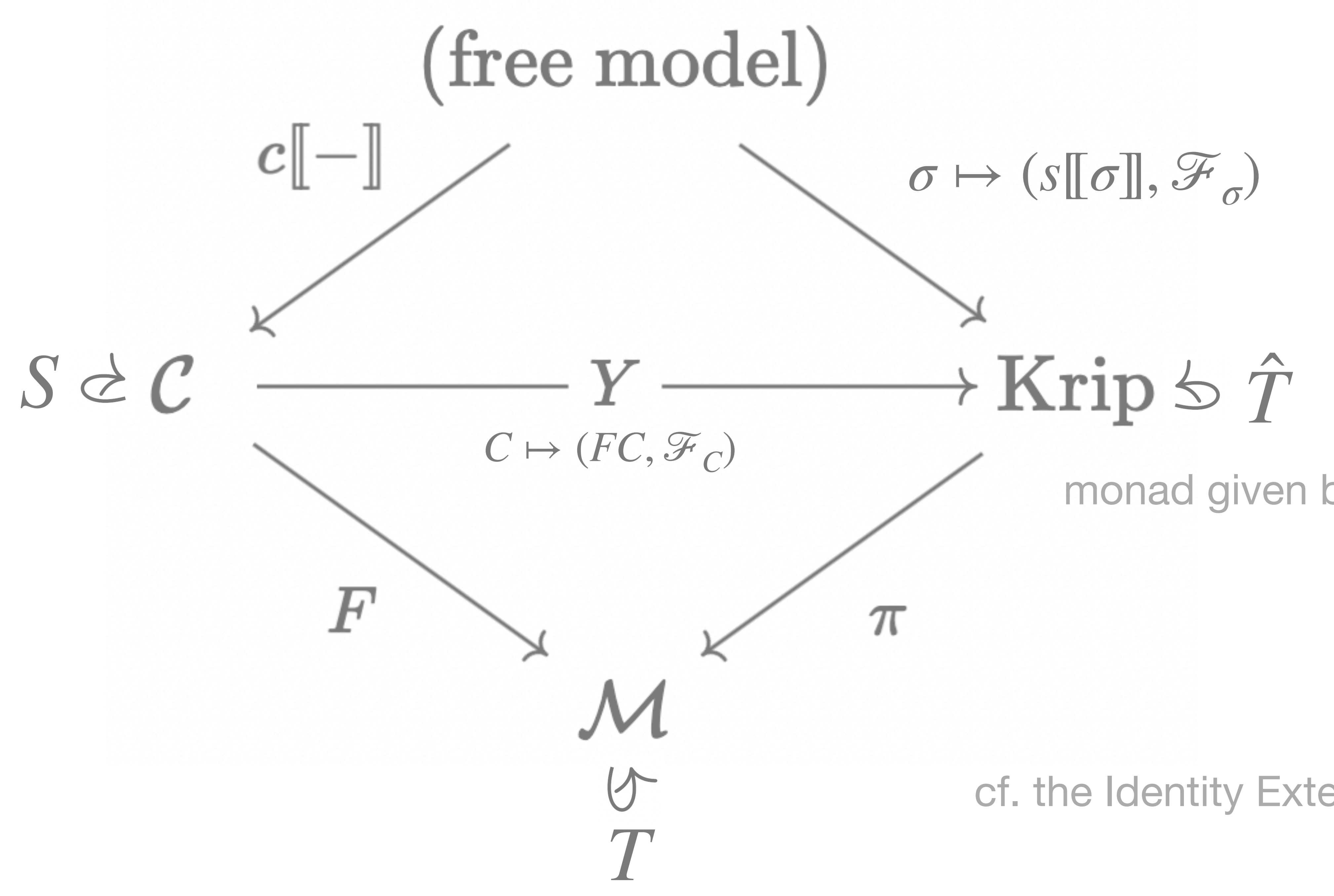
Every morphism of models defines a logical relation:

$$\mathscr{F}_\sigma(\Gamma) := \{Fh \mid h \in \mathscr{C}(c[\![\Gamma]\!], c[\![\sigma]\!])\}$$

take $\mathscr{C}$ the subcategory of $\mathscr{M}$ with:

- objects: $s[\![\sigma]\!]$ for $\sigma \in \mathrm{Type}$
- maps: definable maps

$\Rightarrow$ Def is logical



monad given by ⊤⊤ -lifting

cf. the Identity Extension Lemma

Converse to the Basic Lemma

Every morphism of models defines a hungry logical relation:

- $\mathscr{F}_\sigma(\Gamma) := \{Fh \mid h \in \mathscr{C}(c[\![\Gamma]\!], c[\![\sigma]\!])\}$
- $f : [\![\sigma]\!] \to [\![\tau]\!]$ satisfies $\mathscr{F} \implies f \in \mathscr{F}_\tau(x : \sigma)$

take $\mathscr{C}$ the subcategory of $\mathscr{M}$ with:

- objects: $s[\![\sigma]\!]$ for $\sigma \in \mathrm{Type}$
- maps: definable maps

$\Rightarrow$ Def is logical

$$\begin{tikzcd}
& \text{(free model)} \arrow[dl, "c[\![-]\!]"'] \arrow[dr, "\sigma \mapsto (s[\![\sigma]\!], \mathscr{F}_\sigma)"] & \\
S \circlearrowright \mathcal{C} \arrow[rr, "Y", "C \mapsto (FC, \mathscr{F}_C)"'] \arrow[dr, "F"'] & & \mathbf{Krip} \circlearrowleft \hat{T} \arrow[dl, "\pi"] \\
& \mathcal{M} \circlearrowright T &
\end{tikzcd}$$

monad given by $\top\top$ -lifting

cf. the Identity Extension Lemma

Converse to the Basic Lemma

Every morphism of models defines a hungry logical relation:

- $\mathscr{F}_\sigma(\Gamma) := \{Fh \mid h \in \mathscr{C}(c[\![\Gamma]\!], c[\![\sigma]\!])\}$
- $f : [\![\sigma]\!] \to [\![\tau]\!]$ satisfies $\mathscr{F} \implies f \in \mathscr{F}_\tau(x : \sigma)$

take $\mathscr{C}$ the subcategory of $\mathscr{M}$ with:

- objects: $s[\![\sigma]\!]$ for $\sigma \in \mathrm{Type}$
- maps: definable maps

$\Rightarrow$ Def is logical

$f : [\![\sigma]\!] \to [\![\tau]\!]$ satisfies Def $\implies$ f is definable

$$
\begin{tikzcd}
 & \text{(free model)} \arrow[dl, "c[\![-]\!]"'] \arrow[dr, "\sigma \mapsto (s[\![\sigma]\!], \mathscr{F}_\sigma)"] & \\
S \circlearrowright \mathcal{C} \arrow[rr, "Y", "C \mapsto (FC, \mathscr{F}_C)"'] \arrow[dr, "F"'] & & \mathbf{Krip} \circlearrowleft \hat{T} \arrow[dl, "\pi"] \\
 & \mathcal{M} \circlearrowleft T &
\end{tikzcd}
$$

monad given by $\top\top$-lifting

cf. the Identity Extension Lemma

Converse to the Basic Lemma

Every morphism of models defines a hungry logical relation:

- $\mathscr{F}_\sigma(\Gamma) := \{Fh \mid h \in \mathscr{C}(c[\![\Gamma]\!], c[\![\sigma]\!])\}$
- $f : [\![\sigma]\!] \to [\![\tau]\!]$ satisfies $\mathscr{F} \implies f \in \mathscr{F}_\tau(x : \sigma)$

take $\mathscr{C}$ the subcategory of $\mathscr{M}$ with:

- objects: $s[\![\sigma]\!]$ for $\sigma \in \mathrm{Type}$
- maps: definable maps

$\Rightarrow$ Def is logical

$f : [\![\sigma]\!] \to [\![\tau]\!]$ satisfies Def $\implies$ f is definable

$$
\begin{tikzcd}
 & \text{(free model)} \arrow[dl, "c[\![-]\!]"'] \arrow[dr, "\sigma \mapsto (s[\![\sigma]\!], \mathscr{F}_\sigma)"] & \\
S \circlearrowright \mathcal{C} \arrow[rr, "Y", "C \mapsto (FC, \mathscr{F}_C)"'] \arrow[dr, "F"'] & & \mathbf{Krip} \circlearrowleft \hat{T} \arrow[dl, "\pi"] \\
 & \mathcal{M} \arrow[d, phantom, "\circlearrowleft"] & \\
 & T &
\end{tikzcd}
$$

monad given by $\top\top$ -lifting

cf. the Identity Extension Lemma

$f : [\![\sigma]\!] \to [\![\tau]\!]$ satisfies every logical relation $\iff$ f is definable

Logical relations and categories of concrete relations

Logical relations and categories of concrete relations

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
\mathrm{Conc}(\mathbb{E}) \arrow[loop above, "H\hat{T}j"] \arrow[r, "j" description] \arrow[r, phantom, shift left=4ex, "\top"] \arrow[r, phantom, shift right=4ex, "\top"] & \mathbb{E} \arrow[l, bend right=50, "H"'] \arrow[l, bend left=50, "K"] \arrow[loop above, "\hat{T}"] \arrow[d, "p"] & \\
& \mathcal{M} \arrow[loop below, "T"] & (\text{free model}) \arrow[l, "s[\![-]\!]"] \arrow[ul]
\end{tikzcd}
$$

Logical relations and categories of concrete relations

$$
\begin{tikzcd}
\mathrm{Conc}(\mathbb{E}) \arrow[loop above, "H\hat{T}j"] \arrow[r, "j"] & \mathbb{E} \arrow[loop above, "\hat{T}"] \arrow[l, bend right=40, "H"'] \arrow[l, bend left=40, "K"] \arrow[d, "p"] & \\
& \mathcal{M} \arrow[loop below, "T"] & \text{(free model)} \arrow[l, "s[\![-]\!]"] \arrow[ul, "\text{logical relation}"']
\end{tikzcd}
$$

Logical relations and categories of concrete relations

Category of concrete relations

$\approx$ model with maps satisfying a logical relation

$$
\begin{tikzcd}
\mathrm{Conc}(\mathbb{E}) \arrow[loop above, "H\hat{T}j"] \arrow[r, "j" description] & \mathbb{E} \arrow[loop above, "\hat{T}"] \arrow[l, bend right=40, "H"', ""{name=H, below}] \arrow[l, bend left=40, "K", ""{name=K, above}] \arrow[d, "p"] & \\
 & \mathcal{M} \arrow[loop below, "T"] & (\text{free model}) \arrow[ul, "\text{logical relation}"'] \arrow[l, "s[\![-]\!]"]
\end{tikzcd}
$$

Logical relations and categories of concrete relations

Category of concrete relations

$\approx$ model with maps satisfying a logical relation

$$
\begin{tikzcd}
\mathrm{Conc}(\mathbb{E}) \arrow[loop above, "H\hat{T}j"] \arrow[r, "j"] \arrow[r, phantom, shift left=4, "\top"] \arrow[r, phantom, shift right=4, "\top"] & \mathbb{E} \arrow[l, bend right=40, "H"'] \arrow[l, bend left=40, "K"] \arrow[loop above, "\hat{T}"] \arrow[d, "p"] & \\
& \mathcal{M} \arrow[loop below, "T"] & (\text{free model}) \arrow[l, "s[\![-]\!]"] \arrow[ul, "\text{logical relation}"']
\end{tikzcd}
$$

If every R_σ is concrete:

$$\boxed{\begin{array}{c}\text{model } \mathscr{M} \\ + \\ \text{logical relation } R\end{array}} \rightsquigarrow \boxed{\begin{array}{c}\text{model } \mathrm{Conc}(\mathbb{E}) \text{ of} \\ \text{maps satisfying } R\end{array}}$$

Summing up: logical relations

1. Logical relations can be defined via internal fibrations (at least for STLC, λ_{ml} and λ_c)

2. 2-categorical perspective $\Rightarrow$ a simple characterisation of definability

3. $\mathrm{Conc}(\mathbb{E})$ is a model with maps satisfying some logical relation

   not always obvious what this is from the start!

3: Models with every map definable

(‘full completeness’)

What doesn't work

Logical relations and categories of concrete relations

Every logical relation determines a category of concrete relations

$$
\begin{tikzcd}
\mathrm{Conc}(\mathbb{E}) \arrow[loop above, "H\hat{T}j"] & & \mathbb{E} \arrow[ll, "j"] \arrow[ll, bend right=40, "H"'] \arrow[ll, bend left=40, "K"] \arrow[ll, phantom, shift right=2.5ex, "\top"] \arrow[ll, phantom, shift left=2.5ex, "\top"] \arrow[loop above, "\hat{T}"] \arrow[d, "p"] & \\
& & T \circlearrowright \mathcal{M} & (\text{free model}) \arrow[l, "s[\![-]\!]"] \arrow[ul, "\text{logical relation}"']
\end{tikzcd}
$$

take this to be Def

What doesn't work

Logical relations and categories of concrete relations

Every logical relation determines a category of concrete relations

$$
\begin{tikzcd}[column sep=large, row sep=huge]
\mathrm{Conc}(\mathbb{E}) \arrow[loop above, "H\hat{T}j"] & & \mathbb{E} \arrow[ll, "j" description] \arrow[ll, bend right=40, "H"'] \arrow[ll, bend left=40, "K"] \arrow[loop above, "\hat{T}"] \arrow[d, "p"] & \\
& & T \circlearrowright \mathcal{M} & (\text{free model}) \arrow[l, "s[\![-]\!]"] \arrow[ul, "\text{logical relation}"']
\end{tikzcd}
$$

take this to be Def

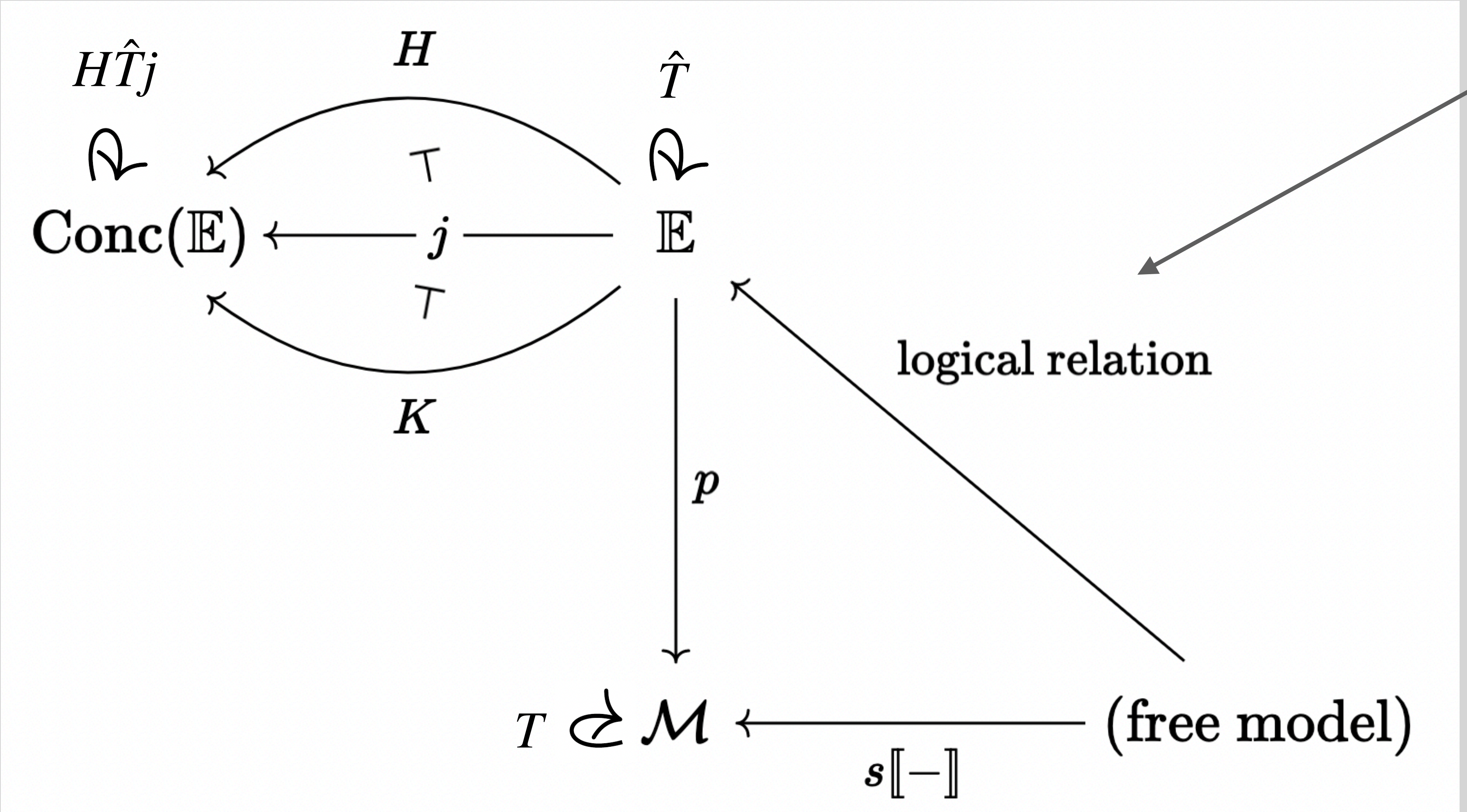
$\Downarrow$

every map in $\mathrm{Conc}(\mathbb{E})$ preserves Def

What doesn't work

Logical relations and categories of concrete relations

Every logical relation determines a category of concrete relations



take this to be Def

$\Downarrow$

every map in $\mathrm{Conc}(\mathbb{E})$ preserves Def

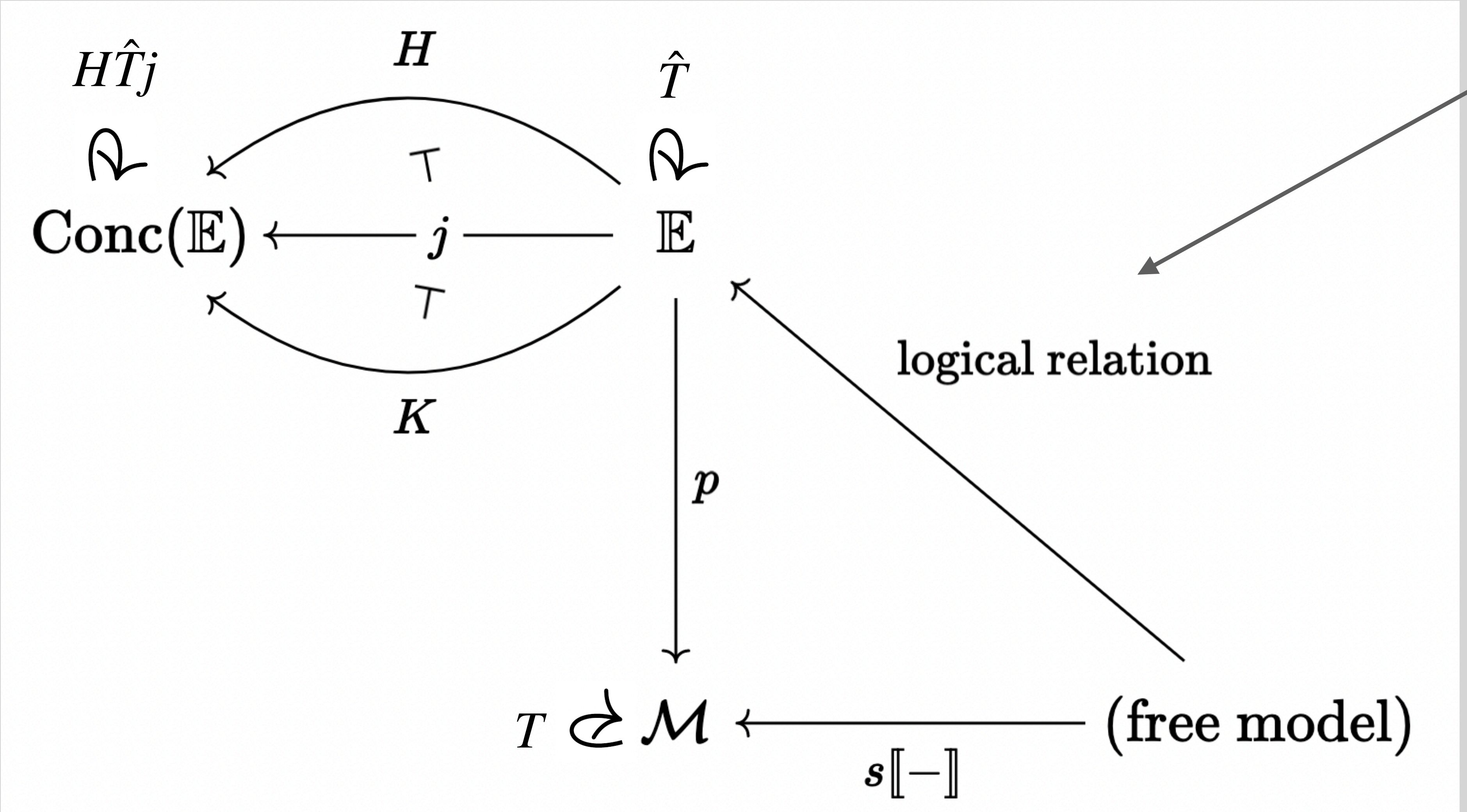
$\Downarrow$

every map in $\mathrm{Conc}(\mathbb{E})$ is definable

What doesn't work

Logical relations and categories of concrete relations

Every logical relation determines a category of concrete relations



take this to be Def

$\Downarrow$

every map in $\mathrm{Conc}(\mathbb{E})$ preserves Def

$\Downarrow$

every map in $\mathrm{Conc}(\mathbb{E})$ is definable

Def for $\mathrm{Conc}(\mathbb{E})$ is not the same as Def for $\mathscr{M}$!

The strategy (cf. O'Hearn & Riecke)

Define a category of concrete relations $\mathrm{OHR}(\mathscr{M})$
with objects $(X, \{R_i \mid i \in I\})$ such that

The strategy (cf. O'Hearn & Riecke)

Define a category of concrete relations $\mathrm{OHR}(\mathscr{M})$
with objects $(X, \{R_i \mid i \in I\})$ such that

for any logical relation $\{L_\sigma \mid \sigma \in \mathrm{Type}\}$ there exists i_0 s.t.

$$\begin{pmatrix} \text{relation at index } i_0 \\ \text{for interpretation of } \sigma \end{pmatrix} = L_\sigma$$

for all $\sigma \in \mathrm{Type}$

ie if $[\![\sigma]\!] = \big(\ldots, \{R_i^\sigma \mid i \in I\}\big)$,
then $R_{i_0}^\sigma = L_\sigma$

The strategy (cf. O'Hearn & Riecke)

Define a category of concrete relations $\mathrm{OHR}(\mathscr{M})$
with objects $(X, \{R_i \mid i \in I\})$ such that

> for any logical relation $\{L_\sigma \mid \sigma \in \mathrm{Type}\}$ there exists i_0 s.t.
>
> $$\begin{pmatrix}\text{relation at index } i_0 \\ \text{for interpretation of } \sigma\end{pmatrix} = L_\sigma$$
>
> for all $\sigma \in \mathrm{Type}$

ie if $[\![\sigma]\!] = \big(\ldots, \{R_i^\sigma \mid i \in I\}\big)$,
then $R_{i_0}^\sigma = L_\sigma$

Then $f : [\![\Gamma]\!] \to H\hat{T}j[\![\sigma]\!]$ in $\mathrm{OHR}(\mathscr{M})$

$\iff$ f is a map in $\mathscr{M}$ preserving every relation R_i

The strategy (cf. O'Hearn & Riecke)

Define a category of concrete relations $\mathrm{OHR}(\mathscr{M})$ with objects $(X, \{R_i \mid i \in I\})$ such that

for any logical relation $\{L_\sigma \mid \sigma \in \mathrm{Type}\}$ there exists i_0 s.t.

$$\begin{pmatrix}\text{relation at index } i_0 \\ \text{for interpretation of } \sigma\end{pmatrix} = L_\sigma$$

for all $\sigma \in \mathrm{Type}$

ie if $[\![\sigma]\!] = \big(\ldots, \{R_i^\sigma \mid i \in I\}\big)$,
then $R_{i_0}^\sigma = L_\sigma$

Then $f : [\![\Gamma]\!] \to H\hat{T}j[\![\sigma]\!]$ in $\mathrm{OHR}(\mathscr{M})$

$\iff$ f is a map in $\mathscr{M}$ preserving every relation R_i

$\implies$ f is a map in $\mathscr{M}$ satisfying L

The strategy (cf. O'Hearn & Riecke)

Define a category of concrete relations $\mathrm{OHR}(\mathscr{M})$ with objects $(X, \{R_i \mid i \in I\})$ such that

for any logical relation $\{L_\sigma \mid \sigma \in \mathrm{Type}\}$ there exists i_0 s.t.

$$\begin{pmatrix}\text{relation at index } i_0\\ \text{for interpretation of } \sigma\end{pmatrix} = L_\sigma$$

for all $\sigma \in \mathrm{Type}$

ie if $[\![\sigma]\!] = \left(\ldots, \{R_i^\sigma \mid i \in I\}\right)$,
then $R_{i_0}^\sigma = L_\sigma$

Then $f : [\![\Gamma]\!] \to H\hat{T}j[\![\sigma]\!]$ in $\mathrm{OHR}(\mathscr{M})$

$\iff$ f is a map in $\mathscr{M}$ preserving every relation R_i

$\implies$ f is a map in $\mathscr{M}$ satisfying L

then:

$\boxed{f \text{ satisfies every logical relation, so is definable}}$

The strategy (cf. O'Hearn & Riecke)

Define a category of concrete relations $\mathrm{OHR}(\mathscr{M})$
with objects $(X, \{R_i \mid i \in I\})$ such that

for any logical relation $\{L_\sigma \mid \sigma \in \mathrm{Type}\}$ there exists i_0 s.t.

$$\begin{pmatrix} \text{relation at index } i_0 \\ \text{for interpretation of } \sigma \end{pmatrix} = L_\sigma$$

for all $\sigma \in \mathrm{Type}$

How do we choose I and $[\![-]\!]$? The intuition:

$$I = \begin{pmatrix} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathscr{M}) \end{pmatrix} \quad , \quad [\![-]\!] \text{ looks up the required relation}$$

How do we choose I and $[\![-]\!]$? The intuition:

$$I = \begin{pmatrix} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathscr{M}) \end{pmatrix} \quad , \quad [\![-]\!] \text{ looks up the required relation}$$

Circular dependencies!

define $\mathrm{OHR}(\mathscr{M})$

How do we choose I and $[\![-]\!]$? The intuition:

$$I = \left(\begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathscr{M}) \end{array} \right) \qquad , \qquad [\![-]\!] \text{ looks up the required relation}$$

Circular dependencies!

$$\text{define } \mathrm{OHR}(\mathscr{M}) \longrightarrow \begin{array}{c} \text{choose } I \text{ so every logical} \\ \text{relation over } \mathrm{OHR}(\mathscr{M}) \text{ appears} \end{array}$$

How do we choose I and $[\![-]\!]$? The intuition:

$$I = \begin{pmatrix} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathscr{M}) \end{pmatrix} \quad , \qquad [\![-]\!] \text{ looks up the required relation}$$

Circular dependencies!

define $\mathrm{OHR}(\mathscr{M})$ → choose I so every logical relation over $\mathrm{OHR}(\mathscr{M})$ appears

→ define logical relation over $\mathrm{OHR}(\mathscr{M})$

How do we choose I and $[\![-]\!]$? The intuition:

$$I = \left(\begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathscr{M}) \end{array} \right) \quad , \quad [\![-]\!] \text{ looks up the required relation}$$

Circular dependencies!

define $\mathrm{OHR}(\mathscr{M})$

choose I so every logical relation over $\mathrm{OHR}(\mathscr{M})$ appears

define logical relation over $\mathrm{OHR}(\mathscr{M})$

How do we choose I and $[\![-]\!]$? The intuition:

$$I = \left(\begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathscr{M}) \end{array} \right) \quad , \qquad [\![-]\!] \text{ looks up the required relation}$$

Circular dependencies!

define $\mathrm{OHR}(\mathscr{M})$ $\longrightarrow$ choose I so every possible relation over $\mathscr{M}$ appears

How do we choose I and $[\![-]\!]$? The intuition:

$$I = \begin{pmatrix} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathscr{M}) \end{pmatrix}, \qquad [\![-]\!] \text{ looks up the required relation}$$

Circular dependencies!

define $\mathrm{OHR}(\mathscr{M})$

choose I so every possible relation over $\mathscr{M}$ appears

identify logical relations over $\mathrm{OHR}(\mathscr{M})$ amongst relations over $\mathscr{M}$

The OHR construction (cf. O'Hearn & Riecke)

Choose I as above, then construct the following category of concrete relations:

$$
\begin{tikzcd}
\mathrm{OHR}(\mathcal{M}) \arrow[r, "j" description] \arrow[loop above, "H\hat{T}j"] \arrow[r, phantom, shift left=3, "\top"] \arrow[r, phantom, shift right=3, "\top"] & \mathcal{K} \arrow[l, bend right=50, "H"'] \arrow[l, bend left=50, "K"] \arrow[r] \arrow[d, "\pi"'] \arrow[loop above, "\hat{T}"] \arrow[dr, phantom, "\lrcorner", very near start] & \prod_{i\in I}\mathrm{Sub}(\widehat{\mathrm{Con}}) \arrow[d, "\prod_i \mathrm{cod}"] \\
& \mathcal{M} \arrow[r, "\langle F_i\rangle_{i\in I}"'] \arrow[loop below, "T"] & \prod_{i\in I}\widehat{\mathrm{Con}}
\end{tikzcd}
$$

Summary

- Categories of concrete relations are a flexible way to 'cut down' models
- Viewed from a general enough perspective, these restrict to maps satisfying a logical relation
- Basic properties of logical relations follow from abstract nonsense
- Combining this theory $\rightsquigarrow$ can construct fully complete models

Future work

- Does the 'internal fibration' view give the right notion in other cases?
- Can the Basic Lemma etc be phrased completely abstractly?
- Universal property for the OHR construction?

Summary

- Categories of concrete relations are a flexible way to ‘cut down’ models
- Viewed from a general enough perspective, these restrict to maps satisfying a logical relation
- Basic properties of logical relations follow from abstract nonsense
- Combining this theory $\rightsquigarrow$ can construct fully complete models

Future work

- Does the ‘internal fibration’ view give the right notion in other cases?
- Can the Basic Lemma etc be phrased completely abstractly?
- Universal property for the OHR construction?