Logical relations for CB(P)V

SCHOOL OF ARTIFICIAL INTELLIGENCE

UNIVERSITY OF EDINBURGH

Memorandum: SAI-RM-4

Date:- October, 1973

Subject: Lambda-definability and logical relations

Author: G.D. Plotkin

jww. Pedro H. Azevedo de Amorim
(Oxford)

Objectives:

1) What is a logical relation?

2) How do we understand log. rels. denotationally?

3) What is a logical relation in the presence of side effects?

4) How does this picture extend to CBPV?

① What is a logical relation?

(a) The high-level idea

A tool for proving (meta) theoretic properties of logics / programming languages

A tool for proving (meta) theoretic properties of logics / programming languages

eg

- definability — Plotkin (1973), Jung-Tiuryn (1993), ...
  - what parts of a model are the interpretations of terms?
- effect simulation — Milne (1974), ..., Katsumata (2013), ...
  - do two models model effect(s) the same way?
- adequacy — Sieber (1992), ...
  - do the denotational and operational semantics agree at base types?

Idea: logical relations are those relations which are "invariant" under all the term-formation operations.

→ builds on the fact every λ-term's denotation is invariant under permutations BUT that there can be uncountably infinitely many such elements in a model.

"Definition": fix a model. A logical relation R is a family of (n-ary) relations $\{ R_\sigma \subseteq [\![\sigma]\!]^n \mid \sigma \in \mathrm{Type} \}$ such that

"Definition": fix a model. A <u>logical relation</u> R is a family of (n-ary) relations $\{ R_\sigma \subseteq [\![\sigma]\!]^n \mid \sigma \in \mathrm{Type} \}$ such that if

$$\frac{(t_i : \tau_i)_{i \in I}}{op(t_1, \dots, t_n) : \tau}$$

is a term formation rule, then:

"Definition": fix a model. A logical relation R is a family of (n-ary) relations $\{ R_\sigma \subseteq [\![\sigma]\!]^n \mid \sigma \in \mathrm{Type} \}$ such that if

$$\frac{(t_i : \tau_i)_{i \in I}}{op(t_1, \dots, t_n) : \tau}$$

is a term formation rule, then:

$$\begin{array}{c}([\![t_i]\!], \dots, [\![t_i]\!]) \in R_{\tau_i} \\ \text{for all } i\end{array} \implies ([\![op(t_1, \dots, t_n)]\!], \dots, [\![op(t_1, \dots, t_n)]\!]) \in R_\tau$$

"<u>Definition</u>" : fix a model. A <u>logical relation</u> R is a family of (n-ary) relations $\{ R_\sigma \subseteq [\![\sigma]\!]^n \mid \sigma \in \mathrm{Type} \}$ such that if

$$\frac{(t_i : \tau_i)_{i \in I}}{op(t_1, \ldots, t_n) : \tau}$$

is a term formation rule, then:

$$([\![t_i]\!], \ldots, [\![t_i]\!]) \in R_{\tau_i} \text{ for all } i \implies ([\![op(t_1, \ldots, t_n)]\!], \ldots, [\![op(t_1, \ldots, t_n)]\!]) \in R_\tau$$

<u>BASIC LEMMA</u> : if $\vdash t : \sigma$ then $([\![t]\!], \ldots, [\![t]\!]) \in R_\sigma$.

(so long as this is true at base types)

How do we use logical relations?

DEFINABILITY: let $\mathcal{M}$ be a model and $x \in \mathcal{M}$.

Is x the interpretation of some term?

DEFINABILITY: let $\mathcal{M}$ be a model and $x \in \mathcal{M}$.

Is x the interpretation of some term?

STS: there $\exists$ some logical relation R and type σ s.t. $(x, \dots, x) \notin R_\sigma$.

DEFINABILITY : let M be a model and $x \in M$.

Is x the interpretation of some term?

STS : there $\exists$ some logical relation R and type σ s.t. $(x, \ldots, x) \notin R_\sigma$.

eg// [Plotkin, Sieber]

- definability of elements in Scott's D_∞ model
- parallel-or $\exists$ not definable in PCF
  - sparked a huge literature on full abstraction!

[Milner '74]

EFFECT SIMULATION: let $\mathcal{M}$ and $\mathcal{N}$ be two models for the same effect. Do they capture "the same" behaviour?

eg// powerset and list for non-determinism. Is it the case that

$$[\![t]\!]^{\mathcal{P}} = \{x_1, \ldots, x_n\} \iff [\![t]\!]^{List} = \left(\begin{array}{c}\text{some permutation of} \\ [x_1, \ldots, x_n]\end{array}\right) ?$$

EFFECT SIMULATION: let M and N be two models for the same effect. Do they capture "the same" behaviour?

eg// powerset and list for non-determinism. Is it the case that

$$[\![t]\!]^{P} = \{x_1, \_, x_n\} \iff [\![t]\!]^{List} = \left(\begin{array}{c}\text{some permutation of} \\ [x_1, \_, x_n]\end{array}\right) ?$$

STRATEGY: define a logical relation R on $M \times N$ saying they have the same behaviour at base types. Then the interpretation of $\vdash t : \sigma$ is a pair $([\![t]\!]^{M}, [\![t]\!]^{N}) \in R_\sigma$ relating the interpretations.

EFFECT SIMULATION

: let M and N be two models for the same effect. Do they capture "the same" behaviour?

eg// powerset and list for non-determinism. Is it the case that

$$[\![t]\!]^{P} = \{x_1, \_\_, x_n\} \iff [\![t]\!]^{List} = \left(\begin{array}{c}\text{some permutation of} \\ [x_1, \_\_, x_n]\end{array}\right) ?$$

STRATEGY: define a logical relation R on $M \times N$ saying they have the same behaviour at base types. Then the interpretation of $\vdash t : \sigma$ is a pair $([\![t]\!]^{M}, [\![t]\!]^{N}) \in R_\sigma$ relating the interpretations.

Cf. also: Friedman's completeness proof for STLC, adequacy proofs, ...

(1) What is a logical relation?

(b) for STLC

STLC = simply-typed λ-calculus

types $\quad \tau ::= \beta \in \mathrm{Base} \mid 1 \mid \tau_1 \times \tau_2 \mid \sigma \to \tau$

terms $\quad t ::= x \mid () \mid \pi_i(t) \mid \langle t_1, t_2 \rangle \mid t\,u \mid \lambda x. t$

equs $\quad \pi_i \langle t_1, t_2 \rangle =_\beta t_i \quad , \quad (\lambda x. t) u =_\beta t[u/x]$

$t =_\eta \langle \pi_1 t, \pi_2 t \rangle \quad , \quad t =_\eta \lambda x. t^x x .$

$() =_\eta t \quad (\text{for } t : 1)$

STLC = simply-typed λ-calculus

types $\quad \tau ::= \beta \in \mathrm{Base} \mid 1 \mid \tau_1 \times \tau_2 \mid \sigma \to \tau$

terms $\quad t ::= x \mid () \mid \pi_i(t) \mid \langle t_1, t_2 \rangle \mid t\,u \mid \lambda x. t$

equs $\quad \pi_i \langle t_1, t_2 \rangle =_\beta t_i \quad , \quad (\lambda x. t) u =_\beta t[u/x]$

$t =_\eta \langle \pi_1 t, \pi_2 t \rangle \quad , \quad t =_\eta \lambda x. t^x x.$

$() =_\eta t \quad (\text{for } t : 1)$

SEMANTIC MODEL:

ccc $\mathbb{C}$ + $s : \mathrm{Base} \longrightarrow \mathbb{C}$ interpreting base types

... for a set-like model:

Def$^{\underline{n}}$: an $\overset{\text{unary}}{\wedge}$ (STLC) logical relation R is a family of relations $\{R_\sigma \subseteq [\![\sigma]\!] \mid \sigma \in Type\}$ s.t.

... for a set-like model:

DEFN: an (unary) (STLC) logical relation R is a family of relations $\{R_\sigma \subseteq [\![\sigma]\!] \mid \sigma \in \mathrm{Type}\}$ s.t.

① $R_1 = \{*\}$

② $(x_1, x_2) \in R_{\sigma_1 \times \sigma_2} \subseteq [\![\sigma_1]\!] \times [\![\sigma_2]\!]$

$$\iff x_i \in R_{\sigma_i} \quad \text{for } i = 1,2$$

ie. $p \in R_{\sigma_1 \times \sigma_2} \iff \pi_i(p) \in R_{\sigma_i}$ for $i = 1,2$

... for a set-like model:

Def$^{\underline{n}}$: an $\overset{\text{unary}}{\wedge}$ (STLC) logical relation R is a family of relations $\{ R_\sigma \subseteq [\![\sigma]\!] \mid \sigma \in \mathrm{Type} \}$ s.t.

① $R_1 = \{ * \}$

② $(x_1, x_2) \in R_{\sigma_1 \times \sigma_2} \subseteq [\![\sigma_1]\!] \times [\![\sigma_2]\!]$

$\iff x_i \in R_{\sigma_i}$ for $i = 1, 2$

ie. $p \in R_{\sigma_1 \times \sigma_2} \iff \pi_i(p) \in R_{\sigma_i}$ for $i = 1, 2$

③ $f \in R_{\sigma \to \tau} \subseteq [\![\sigma]\!] \Rightarrow [\![\tau]\!]$

$\iff \forall x \in R_\sigma \subseteq [\![\sigma]\!].\ f\,x \in R_\tau \subseteq [\![\tau]\!]$

//

... for a CCC $\mathbb{C}$:

DEFN: an $\overset{\text{unary}}{\wedge}$ (STLC) logical relation R is a family of relations $\{R_\sigma \subseteq \mathbb{C}(1, [\![\sigma]\!]) \mid \sigma \in \mathrm{Type}\}$ s.t.

① $R_1 = \{*\}$

② $\langle x_1, x_2 \rangle \in R_{\sigma_1 \times \sigma_2} \subseteq \mathbb{C}(1, [\![\sigma_1]\!] \times [\![\sigma_2]\!])$

$\iff x_i \in R_{\sigma_i}$ for $i = 1,2$

i.e. $p \in R_{\sigma_1 \times \sigma_2} \iff \pi_i(p) \in R_{\sigma_i}$ for $i = 1,2$

③ $f \in R_{\sigma \to \tau} \subseteq \mathbb{C}(1, [\![\sigma]\!] \Rightarrow [\![\tau]\!])$

$\iff \forall x \in R_\sigma \subseteq \mathbb{C}(1, [\![\sigma]\!])$,

$1 \xrightarrow{\langle f, x \rangle} ([\![\sigma]\!] \Rightarrow [\![\tau]\!]) \times [\![\sigma]\!] \xrightarrow{\mathrm{eval}} [\![\tau]\!] \in R_\tau$

DEF$^{\underline{N}}$: an (STLC) logical relation R is a family of relations $\{R_\sigma \subseteq \mathbb{C}(1, [\![\sigma]\!]) \mid \sigma \in \mathrm{Type}\}$ s.t.

unary

① $R_1 = \{*\}$

② $\langle x_1, x_2 \rangle \in R_{\sigma_1 \times \sigma_2} \subseteq \mathbb{C}(1, [\![\sigma_1]\!] \times [\![\sigma_2]\!])$

$\iff x_i \in R_{\sigma_i}$ for $i = 1, 2$

ie. $p \in R_{\sigma_1 \times \sigma_2} \iff \pi_i(p) \in R_{\sigma_i}$ for $i = 1, 2$

③ $f \in R_{\sigma \to \tau} \subseteq \mathbb{C}(1, [\![\sigma]\!] \Rightarrow [\![\tau]\!])$

$\iff \forall x \in R_\sigma \subseteq \mathbb{C}(1, [\![\sigma]\!]).$

$$1 \xrightarrow{\langle f, x \rangle} ([\![\sigma]\!] \Rightarrow [\![\tau]\!]) \times [\![\sigma]\!] \xrightarrow{eval} [\![\tau]\!] \in R_\tau$$

Basic lemma: if R is a logical relation and $\vdash t : \sigma$ then $[\![t]\!] \in R_\sigma$.

Now you have many variants:

- n-ary relations vs unary relations ...
- varying arity = taking account of contexts ...
- adding sum types or constants ...

⋮

Now you have many variants:

- n-ary relations vs unary relations ...
- varying arity = taking account of contexts ...
- adding sum types or constants ...

⋮

There are lots of 'tweaked' versions of logical relations for STLC et al, which solve slightly different problems. (eg. adding sums or other types)

(2) How do we understand logical relations <u>denotationally</u>?

BACK TO STLC IN SET :

given $s : \mathrm{Base} \longrightarrow \mathrm{Set}$ so we get $s[\![t]\!] : [\![\Gamma]\!] \to [\![\sigma]\!]$ for every term $\Gamma \vdash t : \sigma$. How does R_σ come in?

BACK TO STLC IN SET :

given $s : \mathrm{Base} \longrightarrow \mathrm{Set}$ so we get $s[\![t]\!] : [\![\Gamma]\!] \to [\![\sigma]\!]$ for every term $\Gamma \vdash t : \sigma$. How does R_σ come in?

<u>DEF$^{\underline{N}}$</u> : <u>Pred</u> is the category with

- objects : sets X with a subset $\bar{X} \subseteq X$
- maps $(X, \bar{X}) \longrightarrow (Y, \bar{Y})$: functions $f : X \to Y$ which preserve the relation : $x \in \bar{X} \Rightarrow fx \in \bar{Y}$.

//

Pred is a ccc

$$\mathbb{1} := (1, 1)$$

$$(X_1, \bar{X}_1) \times (X_2, \bar{X}_2) := (X_1 \times X_2, \bar{X}_1 \times \bar{X}_2)$$

$$(X, \bar{X}) \Rightarrow (Y, \bar{Y}) := (X \Rightarrow Y, \bar{X} \supset \bar{Y})$$

where $f \in \bar{X} \supset \bar{Y} \Leftrightarrow \forall x \in \bar{X}.\ f\,x \in \bar{Y}$

"functions that preserve the relation"

Pred is a ccc

$$1 := (1, 1)$$

$$(X_1, \overline{X}_1) \times (X_2, \overline{X}_2) := (X_1 \times X_2, \overline{X}_1 \times \overline{X}_2)$$

$$(X, \overline{X}) \Rightarrow (Y, \overline{Y}) := (X \Rightarrow Y, \overline{X} \supset \overline{Y})$$

where $f \in \overline{X} \supset \overline{Y} \Leftrightarrow \forall x \in \overline{X}.\ f\,x \in \overline{Y}$

... and the forgetful functor $p : \mathbf{Pred} \longrightarrow \mathbf{Set}$ strictly preserves this structure

Pred is a ccc

$$1 := (1, 1)$$

$$(X_1, \overline{X}_1) \times (X_2, \overline{X}_2) := (X_1 \times X_2, \overline{X}_1 \times \overline{X}_2)$$

$$(X, \overline{X}) \Rightarrow (Y, \overline{Y}) := (X \Rightarrow Y, \overline{X} \supset \overline{Y})$$

where $f \in \overline{X} \supset \overline{Y} \Leftrightarrow \forall x \in \overline{X}.\ f\,x \in \overline{Y}$

... and the forgetful functor $p : \mathbf{Pred} \longrightarrow \mathbf{Set}$ strictly preserves this structure

NOW WE CAN SEE WHERE LOG. RELS. COME FROM ...

Suppose we pick for each $\beta \in \mathrm{Base}$ a relation $R_\beta \subseteq [\![\beta]\!]$.

This amounts to:

$$\begin{tikzcd} & \mathrm{Pred} \arrow[d, "p"] \\ \mathrm{Base} \arrow[ur, "\hat{s}"] \arrow[r, "s"'] & \mathrm{Set} \end{tikzcd}$$

ccc

strictly preserves ccc

ccc

Suppose we pick for each $\beta \in \mathrm{Base}$ a relation $R_\beta \subseteq [\![\beta]\!]$.

This amounts to:

$$
\begin{tikzcd}
 & \mathrm{Pred} \arrow[d, "p"] & \text{ccc} \\
 & & \text{strictly preserves ccc} \\
\mathrm{Base} \arrow[uur, "\hat{s}"] \arrow[r, "s"'] & \mathrm{Set} & \text{ccc}
\end{tikzcd}
$$

Hence, by induction / initiality, we get

$$
\begin{tikzcd}
 & & \mathrm{Pred} \arrow[d, "p"] \\
\text{syntactic model} & \mathrm{Syn} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathrm{Set}
\end{tikzcd}
$$

Suppose we pick for each $\beta \in \mathrm{Base}$ a relation $R_\beta \subseteq [\![\beta]\!]$.

This amounts to:

$$\begin{tikzcd} & \mathrm{Pred} \arrow[d, "p"] \\ \mathrm{Base} \arrow[ur, "\hat{s}"] \arrow[r, "s"'] & \mathrm{Set} \end{tikzcd}$$

ccc

strictly preserves ccc

ccc

Hence, by induction / initiality, we get

$$\begin{tikzcd} & \mathrm{Pred} \arrow[d, "p"] \\ \mathrm{Syn} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathrm{Set} \end{tikzcd}$$

syntactic model

$\hat{s}[\![\sigma]\!]$ encodes a logical relation:

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

syntactic model

$$
\begin{tikzcd}
 & \mathrm{Pred} \arrow[d, "p"] \\
\mathrm{Syn} \arrow[ur, "\tilde{S}[\![-]\!]"] \arrow[r, "S[\![-]\!]"'] & \mathrm{Set}
\end{tikzcd}
$$

$$\mathbb{1} := (1, 1)$$

$$(X_1, \bar{X}_1) \times (X_2, \bar{X}_2) := (X_1 \times X_2, \bar{X}_1 \times \bar{X}_2)$$

$$(X, \bar{X}) \Rightarrow (Y, \bar{Y}) := (X \Rightarrow Y, \bar{X} \supset \bar{Y})$$

where $f \in \bar{X} \supset \bar{Y} \iff \forall x \in \bar{X}.\ f\,x \in \bar{Y}$

syntactic model

$$\begin{tikzcd} & \mathrm{Pred} \arrow[d, "p"] \\ \mathrm{Syn} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathrm{Set} \end{tikzcd}$$

$$\mathbf{1} := (\mathbf{1}, \mathbf{1})$$

$$(X_1, \bar{X}_1) \times (X_2, \bar{X}_2) := (X_1 \times X_2, \bar{X}_1 \times \bar{X}_2)$$

$$(X, \bar{X}) \Rightarrow (Y, \bar{Y}) := (X \Rightarrow Y, \bar{X} \supset \bar{Y})$$

where $f \in \bar{X} \supset \bar{Y} \iff \forall x \in \bar{X}.\ f\, x \in \bar{Y}$

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

$$\hat{s}[\![\mathbf{1}]\!] = (\mathbf{1}, \{*\}) = (\mathbf{1}, \underline{R_{\mathbf{1}}})$$

syntactic model

$$\begin{tikzcd} & \mathbf{Pred} \arrow[d, "p"] \\ \mathbf{Syn} \arrow[ur, "\hat{s}[\![-]\!]"] \arrow[r, "s[\![-]\!]"'] & \mathbf{Set} \end{tikzcd}$$

$$\mathbb{1} := (1, 1)$$

$$(X_1, \bar{X}_1) \times (X_2, \bar{X}_2) := (X_1 \times X_2, \bar{X}_1 \times \bar{X}_2)$$

$$(X, \bar{X}) \Rightarrow (Y, \bar{Y}) := (X \Rightarrow Y, \bar{X} \supset \bar{Y})$$

where $f \in \bar{X} \supset \bar{Y} \Leftrightarrow \forall x \in \bar{X}.\ f\,x \in \bar{Y}$

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

$$\hat{s}[\![\mathbb{1}]\!] = (1, \{*\}) = (1, \underline{R_{\mathbb{1}}})$$

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

syntactic model

$$\begin{tikzcd} & \mathrm{Pred} \arrow[d, "p"] \\ \mathrm{Syn} \arrow[ur, "\hat{s}\llbracket - \rrbracket"] \arrow[r, "s\llbracket - \rrbracket"'] & \mathrm{Set} \end{tikzcd}$$

$$1 := (1, 1)$$

$$(X_1, \bar{X}_1) \times (X_2, \bar{X}_2) := (X_1 \times X_2, \bar{X}_1 \times \bar{X}_2)$$

$$(X, \bar{X}) \Rightarrow (Y, \bar{Y}) := (X \Rightarrow Y, \bar{X} \supset \bar{Y})$$

where $f \in \bar{X} \supset \bar{Y} \iff \forall x \in \bar{X}.\ f\,x \in \bar{Y}$

$$\hat{s}\llbracket \beta \rrbracket = (s\llbracket \beta \rrbracket, \underline{R_\beta})$$

$$\hat{s}\llbracket 1 \rrbracket = (1, \{*\}) = (1, \underline{R_1})$$

$$\hat{s}\llbracket \sigma_1 \times \sigma_2 \rrbracket = \hat{s}\llbracket \sigma_1 \rrbracket \times \hat{s}\llbracket \sigma_2 \rrbracket = (s\llbracket \sigma_1 \rrbracket, R_{\sigma_1}) \times (s\llbracket \sigma_2 \rrbracket, R_{\sigma_2}) = (s\llbracket \sigma_1 \rrbracket \times s\llbracket \sigma_2 \rrbracket, \underline{R_{\sigma_1} \times R_{\sigma_2}})$$

$$\hat{s}\llbracket \sigma \to \tau \rrbracket = \hat{s}\llbracket \sigma \rrbracket \Rightarrow \hat{s}\llbracket \tau \rrbracket = (s\llbracket \sigma \rrbracket, R_\sigma) \Rightarrow (s\llbracket \tau \rrbracket, R_\tau) = (s\llbracket \sigma \rrbracket \Rightarrow s\llbracket \tau \rrbracket, \underline{R_\sigma \supset R_\tau})$$

SLOGAN: to give an STLC logical relation for a model $(\mathbb{C}, s)$ is to give

[Reynolds, Ma]

1. a a ccc $\mathbb{E}$ of "relations"

2. a functor $p : \mathbb{E} \longrightarrow \mathbb{C}$ strictly preserving the ccc structure

3. an interpretation $\hat{s}$ in $\mathbb{E}$ s.t.

$$\begin{tikzcd}
 & \mathbb{E} \arrow[d, "p"] \\
\mathrm{Base} \arrow[ur, "\hat{s}"] \arrow[r, "s"'] & \mathbb{C}
\end{tikzcd}$$

SLOGAN: to give an STLC logical relation for a Model $(\mathbb{C}, s)$ is to give

[Reynolds, Ma]

① a a ccc $\mathbb{E}$ of "relations"

② a functor $p : \mathbb{E} \longrightarrow \mathbb{C}$ strictly preserving the ccc structure

③ an interpretation $\hat{s}$ in $\mathbb{E}$ s.t.

$$\begin{tikzcd} & \mathbb{E} \arrow[d, "p"] \\ \mathrm{Base} \arrow[ur, "\hat{s}"] \arrow[r, "s"'] & \mathbb{C} \end{tikzcd}$$

... then R_σ arises via $\hat{s}[\![\sigma]\!]$ in $\mathbb{E}$.

SLOGAN: to give an STLC logical relation for a Model $(\mathbb{C}, s)$ is to give

[Reynolds, Ma]

what is this? [① a a ccc $\mathbb{E}$ of "relations"

② a functor $p : \mathbb{E} \longrightarrow \mathbb{C}$ strictly preserving the ccc structure

③ an interpretation $\hat{s}$ in $\mathbb{E}$ s.t.

$$\begin{tikzcd} & \mathbb{E} \arrow[d, "p"] \\ \mathrm{Base} \arrow[ur, "\hat{s}"] \arrow[r, "s"'] & \mathbb{C} \end{tikzcd}$$

... then R_σ arises via $\hat{s}[\![\sigma]\!]$ in $\mathbb{E}$.

from this perspective,

logical relations is about building new, "relational", models over existing models

RELATIONS MODELS

- how can we spot them?
- how can we reason about them?
- what facts do we know hold for any such model?

RELATIONS MODELS

- how can we spot them?
- how can we reason about them?
- what facts do we know hold for any such model?

↳ as "**fibrations for logical relations**"

$P : \mathrm{Pred} \longrightarrow \mathrm{Set}$ has a special property:

given $\overline{Y} \subseteq Y$ and $f : X \to Y$, there is a canonical way to "pull back" the predicate $\overline{Y}$ to a predicate on X, s.t. f lifts to a map in Pred:

p : Pred $\longrightarrow$ Set has a special property:

given $\bar{Y} \subseteq Y$ and $f: X \to Y$, there is a canonical way to "pull back" the predicate $\bar{Y}$ to a predicate on X, s.t. f lifts to a map in Pred:

$$\begin{array}{ccc} & (Y, \bar{Y}) & \\ & & \\ X \xrightarrow[f]{} & Y = p(Y, \bar{Y}) & \end{array} \qquad \begin{array}{c} \text{Pred} \\ \downarrow p \\ \text{Set} \end{array}$$

p : Pred $\longrightarrow$ Set has a special property:

given $\bar{Y} \subseteq Y$ and $f : X \to Y$, there is a canonical way to "pull back" the predicate $\bar{Y}$ to a predicate on X, s.t. f lifts to a map in Pred:

$$
\begin{tikzcd}
(X, ??) \arrow[r, dashed, "\hat{f}"'] & (Y, \bar{Y}) \\
X \arrow[r, "f"'] & Y = p(Y, \bar{Y})
\end{tikzcd}
\qquad
\begin{tikzcd}
\text{Pred} \arrow[d, "p"] \\
\text{Set}
\end{tikzcd}
$$

$p : \mathrm{Pred} \longrightarrow \mathrm{Set}$ has a special property:

given $\bar{Y} \subseteq Y$ and $f : X \to Y$, there is a canonical way to "pull back" the predicate $\bar{Y}$ to a predicate on X, s.t. f lifts to a map in Pred:

$$\begin{tikzcd}
(X, ??) \arrow[r, dashed, "\hat{f}"'] & (Y, \bar{Y}) & & \mathrm{Pred} \arrow[d, "p"] \\
X \arrow[r, "f"'] & Y = p(Y, \bar{Y}) & & \mathrm{Set}
\end{tikzcd}$$

$f^{-1}[\bar{Y}] \subseteq X$

DEF$^{\underline{N}}$: a (Grothendieck) <u>fibration</u> is a functor $p : \mathbb{E} \to \mathbb{B}$ such that for every $f : A \longrightarrow pY$ in $\mathbb{B}$ there is an object $\hat{A} \in \mathbb{E}$ and a map $\hat{f} : \hat{A} \longrightarrow Y$ s.t.

- $p(\hat{A}) = A$
- $p(\hat{f}) = f$
- a universal property holds

DEF$^{\underline{N}}$: a (Grothendieck) <u>fibration</u> is a functor $p : \mathbb{E} \to \mathbb{B}$ such that for every $f : A \longrightarrow pY$ in $\mathbb{B}$ there is an object $\hat{A} \in \mathbb{E}$ and a map $\hat{f} : \hat{A} \longrightarrow Y$ s.t.

- $p(\hat{A}) = A$
- $p(\hat{f}) = f$
- a universal property holds

PICTORIALLY:

$$
\begin{tikzcd}
\hat{A} \arrow[r, dashed, "\hat{f}"'] & Y & & \mathbb{E} \arrow[d, "p"] \\
A \arrow[r, "f"'] & pY & & \mathbb{B}
\end{tikzcd}
$$

EXAMPLES

- $p : \mathrm{Pred}_n \longrightarrow \mathrm{Set}$ n-ary predicates over Set

  ↳ more generally $\mathrm{Sub}_n(\mathbb{C}) \longrightarrow \mathbb{C}$ n-ary subobjects over $\mathbb{C}$ for nice enough $\mathbb{C}$

- $\mathrm{cod} : \mathbb{C}^{\to} \longrightarrow \mathbb{C}$ the codomain fibration

  <u>obj</u> : $f : X \longrightarrow Y$ in $\mathbb{C}$

  <u>maps</u> : squares
$$\begin{tikzcd} X \arrow[r] \arrow[d, "f"'] \arrow[dr, phantom, "/\!/"] & X' \arrow[d, "f'"] \\ Y \arrow[r] & Y' \end{tikzcd}$$

A RECIPE :

C

ccc

A RECIPE:

ccc

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

fibration strictly preserving ccc structure

ccc

ccc

Product preserving

A RECIPE :

fibration strictly preserving ccc structure

$$\begin{tikzcd} \overset{ccc}{\hat{\mathbb{C}}} \arrow[r] \arrow[d, "q"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E}^{ccc} \arrow[d, "p"] \\ \underset{ccc}{\mathbb{C}} \arrow[r, "F"'] & \mathbb{B}_{ccc} \end{tikzcd}$$

fibration strictly preserving ccc structure

F product preserving

A RECIPE :

obj : $(C \in \mathbb{C}, X \in \mathbb{E})$ st. $FC = pX$,

i.e. a $\mathbb{C}$-object and a 'relation' on it

$$\begin{tikzcd}
\hat{\mathbb{C}}^{\text{ccc}} \arrow[r] \arrow[d, "q"'] \arrow[dr, phantom, "\lrcorner" very near start] & \mathbb{E}^{\text{ccc}} \arrow[d, "p"] \\
\mathbb{C}_{\text{ccc}} \arrow[r, "F"'] & \mathbb{B}_{\text{ccc}}
\end{tikzcd}$$

fibration strictly preserving ccc structure

F product preserving

fibration strictly preserving ccc structure

EXAMPLES

obj: $(X, R \subseteq X \times X)$

①

$$\begin{tikzcd}
\mathrm{BinPred} \arrow[r] \arrow[d, "\dashv"] & \mathrm{Pred} \arrow[d, "p"] \\
\mathrm{Set} \arrow[r, "X \mapsto X \times X"'] & \mathrm{Set}
\end{tikzcd}$$

obj: $(X, Y, R \subseteq X \times Y)$

②

$$\begin{tikzcd}
\hat{C} \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Pred} \arrow[d, "p"] \\
\mathrm{Set} \times \mathrm{Set} \arrow[r, "\times"'] & \mathrm{Set}
\end{tikzcd}$$

can start to relate models

EXAMPLES

obj: $(X, R \subseteq X \times X)$

①
$$\begin{tikzcd} \mathrm{BinPred} \arrow[r] \arrow[d, "\dashv"'] & \mathrm{Pred} \arrow[d, "p"] \\ \mathrm{Set} \arrow[r, "X \mapsto X \times X"'] & \mathrm{Set} \end{tikzcd}$$

② obj: $(X, Y, R \subseteq X \times Y)$

$$\begin{tikzcd} \hat{\mathbb{C}} \arrow[r] \arrow[d, "\dashv"'] & \mathrm{Pred} \arrow[d, "p"] \\ \mathrm{Set} \times \mathrm{Set} \arrow[r, "\times"'] & \mathrm{Set} \end{tikzcd}$$

can start to <u>relate</u> models

③
$$\begin{tikzcd} \hat{\mathbb{C}} \arrow[r] \arrow[d, "q"', "\dashv"] & \mathrm{Sub}[\mathrm{Ctx}, \mathrm{Set}] \arrow[d, "p"] \\ \mathbb{C} \arrow[r, "X \mapsto \mathbb{C}(\llbracket - \rrbracket, X)"'] & [\mathrm{Ctx}, \mathrm{Set}] \end{tikzcd}$$

Kripke relations of varying arity

obj: $(X \in \mathbb{C},\ R \rightarrowtail \mathbb{C}(\llbracket - \rrbracket, X))$

ie. for every context Γ a subset

$R(\Gamma) \subseteq \mathbb{C}(\llbracket \Gamma \rrbracket, X)$,

compatible with renamings

SUMMING UP

: we have a nice denotational account of STLC logical relations

SUMMING UP

: we have a nice denotational account of STLC logical relations

① denotationally, logical relations is about constructing refined models over existing ones

SUMMING UP

: we have a nice denotational account of STLC logical relations

① denotationally, logical relations is about constructing refined models over existing ones

② such models can be identified as fibrations which strictly preserve the ccc structure — ie. the model

SUMMING UP

: we have a nice denotational account of STLC logical relations

1. denotationally, logical relations is about constructing refined models over existing ones

2. such models can be identified as fibrations which strictly preserve the ccc structure — ie. the model

3. the general theory gives us a useful framework for finding such fibrations (= relations models)

SUMMING UP

: we have a nice denotational account of STLC logical relations

① denotationally, logical relations is about constructing refined models over existing ones

② such models can be identified as fibrations which strictly preserve the ccc structure — ie. the model

③ the general theory gives us a useful framework for finding such fibrations (= relations models)

DEF$\underline{\text{N}}$: an $\underline{\text{STLC fibration}}$ is a fibration that strictly preserves ccc structure $\approx$ a logical relation

③ What is a logical relation in the presence of side effects?

└ spoiler : it will be a fibration which strictly preserves the model structure

SIDE EFFECTS

STLC programs are simple: they always terminate, never interact with the world in interesting ways

SIDE EFFECTS

STLC programs are simple: they always terminate, never interact with the world in interesting ways

actual programs have things like

- exceptions
- non-determinism
- state
- user input
- output / printing
- probabilistic behaviours

⋮

Side Effects

STLC programs are simple: they always terminate, never interact with the world in interesting ways

actual programs have things like

structure of these programs captured by monads

- exceptions
- non-determinism
- state
- user input
- output / printing
- probabilistic behaviours

⋮

Monadic Metalanguage

Extend STLC with

types $\tau ::= \ldots \mid T\tau$

run u, bind it to x, then run t.
Haskell notation:
$t \gg= x$

terms $t ::= \ldots \mid \text{return } t \mid \text{let } x = u \text{ in } t$

a pure program is trivially effectful

equations $\ldots$ $\text{let } x = (\text{return } u) \text{ in } t =_\beta t[u/x]$ $\ldots$

Example Monads

- List or powerset for non-determinism
- Maybe $= (-) + 1$ for non-termination
- Exception $= (-) + E$ for a set E of exceptions
- Writer $= (-) \times C^*$ for C a set of characters
- $K_R := (- \Rightarrow R) \Rightarrow R$ for continuations

SEMANTICS : ccc $\mathcal{C}$ + a (strong) monad T

$$[\![T\sigma]\!] := T[\![\sigma]\!]$$

$$[\![\text{return } t]\!] := [\![\Gamma]\!] \xrightarrow{[\![t]\!]} [\![\sigma]\!] \xrightarrow{\eta} T[\![\sigma]\!]$$

SEMANTICS : ccc $\mathcal{C}$ + a (strong) monad T

$$[\![T\sigma]\!] := T[\![\sigma]\!]$$

$$[\![\text{return } t]\!] := [\![\Gamma]\!] \xrightarrow{[\![t]\!]} [\![\sigma]\!] \xrightarrow{\eta} T[\![\sigma]\!]$$

eg//

for powerset : $[\![\text{return } t]\!](\gamma) = \{[\![t]\!](\gamma)\}$

for Maybe : $[\![\text{return } t]\!](\gamma) = \text{inl } ([\![t]\!]\,\gamma) \in [\![\sigma]\!] + 1$

for Writer : $[\![\text{return } t]\!](\gamma) = ([\![t]\!](\gamma), \varepsilon)$

Semantics : CCC $\mathcal{C}$ + a (strong) Monad T

$$[\![T\sigma]\!] := T[\![\sigma]\!]$$

$$[\![\text{return } t]\!] := [\![\Gamma]\!] \xrightarrow{[\![t]\!]} [\![\sigma]\!] \xrightarrow{\eta} T[\![\sigma]\!]$$

$$[\![\text{let } x = u \text{ in } t]\!] := [\![\Gamma]\!] \xrightarrow{\langle id, [\![u]\!]\rangle} [\![\Gamma]\!] \times T[\![\sigma]\!] \xrightarrow{\text{strength}} T([\![\Gamma]\!] \times [\![\sigma]\!]) \xrightarrow{T[\![t]\!]} T^2[\![\tau]\!] \xrightarrow{\mu} T[\![\tau]\!]$$

SEMANTICS : ccc $\mathcal{C}$ + a (strong) monad T

$$[\![T\sigma]\!] := T[\![\sigma]\!]$$

$$[\![\text{return } t]\!] := [\![\Gamma]\!] \xrightarrow{[\![t]\!]} [\![\sigma]\!] \xrightarrow{\eta} T[\![\sigma]\!]$$

$$[\![\text{let } x = u \text{ in } t]\!] := [\![\Gamma]\!] \xrightarrow{\langle id, [\![u]\!]\rangle} [\![\Gamma]\!] \times T[\![\sigma]\!] \xrightarrow{\text{strength}} T([\![\Gamma]\!] \times [\![\sigma]\!]) \xrightarrow{T[\![t]\!]} T^2[\![\tau]\!] \xrightarrow{\mu} T[\![\tau]\!]$$

eg. for powerset : $[\![u]\!](\gamma) \in \mathcal{P}[\![\sigma]\!]$ and $[\![t]\!] : [\![\Gamma]\!] \times [\![\sigma]\!] \longrightarrow \mathcal{P}[\![\tau]\!]$, so

$$[\![\text{let } x = u \text{ in } t]\!](\gamma) = \text{let } [\![u]\!](\gamma) = S \subseteq \mathcal{P}[\![\sigma]\!] \text{ in } \bigcup_{s \in S} [\![t]\!](\gamma, s)$$

for Writer:

$$[\![\text{let } x = u \text{ in } t]\!](\gamma) = \text{let } [\![u]\!](\gamma) = (x, w) \in [\![\sigma]\!] \times C^* \text{ in}$$
$$\text{let } [\![t]\!](\gamma, x) = (y, w') \in [\![\tau]\!] \times C^* \text{ in}$$
$$(y, ww')$$

strict

MAPS OF λ_{ml} MODELS

$$\begin{tikzcd}
\text{ccc + strong monad} \quad (\mathbb{E}, \hat{T}) \arrow[d, "p"] \\
\text{ccc + strong monad} \quad (\mathbb{B}, T)
\end{tikzcd}$$

strict

MAPS OF λ_{ml} MODELS

$$
\begin{tikzcd}
\text{ccc + strong monad} \quad (\mathbb{E}, \hat{T}) \arrow[d, "p"] \\
\text{ccc + strong monad} \quad (\mathbb{B}, T)
\end{tikzcd}
$$

p preserves ccc structure
+ monad structure on the nose:

$p(\hat{T}X) = T(pX)$

$p(\hat{\mu}_X) = \mu_{pX}$ $\qquad$ $p(\hat{st}_{X,Y}) = st_{pX,pY}$

$p(\hat{\eta}_X) = \eta_{pX}$

EXAMPLE:

1) if $\mathbb{E}$, $\mathbb{B}$ are cccs and $p : \mathbb{E} \longrightarrow \mathbb{B}$ preserves ccc-structure, $(\mathbb{E}, (- \Rightarrow R) \Rightarrow R) \xrightarrow{p} (\mathbb{B}, (- \Rightarrow pR) \Rightarrow pR)$

EXAMPLE:

1) if $\mathbb{E}, \mathbb{B}$ are cccs and $p : \mathbb{E} \to \mathbb{B}$ preserves ccc-structure, $(\mathbb{E}, (- \Rightarrow R) \Rightarrow R) \overset{p}{\longmapsto} (\mathbb{B}, (- \Rightarrow pR) \Rightarrow pR)$

2) if $C \in \mathrm{Set}$ is a set of characters and $\bar{C} \subseteq C$ then $\bar{C}^* \overset{\text{submonoid}}{\rightarrowtail} C^*$ so $(-) \times (C^*, \bar{C}^*) : \mathrm{Pred} \to \mathrm{Pred}$

$(X, \bar{X}) \longmapsto (X \times C^*, \bar{X} \times \bar{C}^*)$

and

$$(\mathrm{Pred}, (-) \times (C^*, \bar{C}^*)) \longrightarrow (\mathrm{Set}, (-) \times C^*)$$

LOGICAL RELATIONS FOR λ_{ml}

Take the STLC picture...

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

$\mathbb{E}$: ccc

p: fibration strictly preserving ccc structure

$\mathbb{B}$: ccc

$\mathbb{C}$: ccc: model of interest

F: product-preserving

LOGICAL RELATIONS FOR λ_{ml}

Take the STLC picture...

$$
\begin{tikzcd}
\hat{\mathbb{C}} \arrow[r, dashed] \arrow[d, dashed, "q"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\
\mathbb{C} \arrow[r, "F"'] & \mathbb{B}
\end{tikzcd}
$$

ccc

ccc

fibration strictly preserving ccc structure

fibration strictly preserving ccc structure

ccc

ccc : model of interp

product-preserving

LOGICAL RELATIONS FOR λ_{ml}

Take the STLC picture...

$$
\begin{tikzcd}
 & \hat{\mathbb{C}} \arrow[r, dashed] \arrow[d, dashed, "q"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[d, "p"] \\
\mathrm{Syn} \arrow[ur, bend left, "\hat{S}[\![-]\!]"] \arrow[r, "S[\![-]\!]"'] & \mathbb{C} \arrow[r, "F"'] & \mathbb{B}
\end{tikzcd}
$$

ccc

ccc

fibration strictly preserving ccc structure

fibration strictly preserving ccc structure

ccc

ccc : model of interpret

product-preserving

$$\hat{S}[\![\sigma]\!] = (S[\![\sigma]\!], R_\sigma)$$

Logical Relations for λ_{ml}

Take the STLC picture... and add monads

$$\begin{tikzcd}
 & & \mathbb{E} \arrow[loop above, "\hat{T}"] \arrow[d, "p"] \\
\mathbb{C} \arrow[loop below, "S"] \arrow[rr, "F"'] & & \mathbb{B} \arrow[loop right, "T"]
\end{tikzcd}$$

ccc + strong monad

fibration strictly preserving ccc structure + monad structure

ccc + strong monad

ccc: model of interest + strong monad

product-preserving + monad-preserving

LOGICAL RELATIONS FOR λ_{ml}

Take the STLC picture... and add monads

$$
\begin{tikzcd}
\hat{\mathbb{C}} \arrow[loop above, "\hat{S}"] \arrow[r, dashed] \arrow[d, dashed, "q"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E} \arrow[loop above, "\hat{T}"] \arrow[d, "p"] \\
\mathbb{C} \arrow[loop below, "S"] \arrow[r, "F"'] & \mathbb{B} \arrow[loop right, "T"]
\end{tikzcd}
$$

$\mathbb{E}$: ccc + strong monad

q: fibration strictly preserving ccc structure + strong monad

p: fibration strictly preserving ccc structure + monad structure

$\mathbb{B}$: ccc + strong monad

$\mathbb{C}$: ccc: model of interest + strong monad

F: product-preserving + monad-preserving

LOGICAL RELATIONS FOR λ_{ml}

Take the STLC picture... and add monads

$$
\begin{tikzcd}
 & & \hat{\mathbb{C}} \arrow[loop above, "\hat{S}"] \arrow[rr, dashed] \arrow[d, dashed, "q"'] \arrow[dr, phantom, "\lrcorner", very near start] & & \mathbb{E} \arrow[loop above, "\hat{T}"] \arrow[d, "p"] \\
\mathrm{Syn} \arrow[urr, bend left, "\hat{S}\mathbb{C}\text{-}D"] \arrow[rr, "S\mathbb{C}\text{-}D"'] & & \mathbb{C} \arrow[loop below, "S"'] \arrow[rr, "F"'] & & \mathbb{B} \arrow[loop above, "T"]
\end{tikzcd}
$$

ccc

strong ccc + monad

fibration strictly preserving ccc structure + strong monad

fibration strictly preserving ccc structure + monad structure

ccc : model of interest + strong monad

product-preserving + monad-preserving

ccc + strong monad

Logical Relations for λ_{ml}

Take the STLC picture... and add monads

$$\hat{S}(S[\![\sigma]\!], R_\sigma) = (S(S[\![\sigma]\!]), R_{T\sigma})$$

$$\begin{tikzcd}
 & \hat{\mathbb{C}} \arrow[loop above, "\hat{S}"] \arrow[rr, dashed] \arrow[d, dashed, "q"'] \arrow[dr, phantom, "\lrcorner", very near start] & & \mathbb{E} \arrow[loop above, "\hat{T}"] \arrow[d, "p"] \\
\mathrm{Syn} \arrow[ur, bend left, "\hat{S}[\![-]\!]"] \arrow[r, "S[\![-]\!]"] & \mathbb{C} \arrow[loop below, "S"] \arrow[rr, "F"] & & \mathbb{B} \arrow[loop right, "T"]
\end{tikzcd}$$

$\hat{\mathbb{C}}$: ccc

q: fibration strictly preserving ccc structure + strong monad

$\mathbb{E}$: ccc + strong monad

p: fibration strictly preserving ccc structure + monad structure

$\mathbb{C}$: ccc: model of interest + strong monad

F: product-preserving + monad-preserving

$\mathbb{B}$: ccc + strong monad

Prop$^{\underline{n}}$. [Katsumata, Katsumata - Kammar - S.]

Suppose you have

$$
\begin{tikzcd}
 & \mathbb{E}_{ccc} \circlearrowleft \hat{T} \arrow[d, "p"', "\text{ccc + monad preserving}"] \\
S \circlearrowright \underset{ccc}{\mathbb{C}} \arrow[r, "F"', "\text{product preserving}" near end, swap] & \underset{ccc}{\mathbb{B}} \circlearrowleft T
\end{tikzcd}
$$

$+ \quad \lambda : FS \Rightarrow ST$ a map of strong monads

PROP$^{\underline{N}}$. [Katsumata, Katsumata - Kammar - S.]

Suppose you have

$$\begin{tikzcd} & & \mathbb{E} \circlearrowleft \hat{T} \ \text{ccc} \arrow[d, "P", "\text{ccc + monad preserving}"'] \\ S \circlearrowleft \underset{\text{ccc}}{\mathbb{C}} \arrow[rr, "F"', "\text{product preserving}"] & & \underset{\text{ccc}}{\mathbb{B}} \circlearrowleft T \end{tikzcd}$$

+ $\lambda : FS \Rightarrow ST$ a map of strong monads

Then you get a universal model $(\hat{\mathbb{C}}, \hat{S})$

PROP$^{\underline{n}}$. [Katsumata, Katsumata - Kammar - S.]

Suppose you have

$$
\begin{tikzcd}
\hat{\mathbb{C}} \arrow[loop above, "\hat{S}"] \arrow[rrr] \arrow[d, "q"'] \arrow[dr, phantom, very near start, "\lrcorner"] & & & \mathbb{E}_{ccc} \arrow[loop above, "\hat{T}"] \arrow[d, "p", "\substack{ccc + \text{monad} \\ \text{preserving}}"'] \\
S \circlearrowright \underset{ccc}{\mathbb{C}} \arrow[rrr, "F"', "\substack{\text{product} \\ \text{preserving}}"] & & & \underset{ccc}{\mathbb{B}} \circlearrowleft T
\end{tikzcd}
$$

$+ \quad \lambda : FS \Rightarrow ST$ a map of strong monads

Then you get a universal model $(\hat{\mathbb{C}}, \hat{S})$ as shown.

Prop$^{\underline{n}}$. [Katsumata, Katsumata - Kammar - S.]

Suppose you have

$$\begin{tikzcd}
\hat{S} \circlearrowright \hat{\mathbb{C}} \arrow[r] \arrow[d, "q"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E}_{ccc} \circlearrowleft \hat{T} \arrow[d, "p", "\text{ccc + monad preserving}"'] \\
S \circlearrowright \underset{ccc}{\mathbb{C}} \arrow[r, "F"', "\text{product preserving}"] & \underset{ccc}{\mathbb{B}} \circlearrowleft T
\end{tikzcd}$$

this is a fibration!

$+ \quad \lambda : FS \Rightarrow ST$ a map of strong monads

Then you get a universal model $(\hat{\mathbb{C}}, \hat{S})$ as shown.

DEFINING A LOGICAL RELATION

$$
\begin{tikzcd}
 & \hat{\mathbb{C}} \arrow[loop above, "\hat{S}"] \arrow[rr] \arrow[d, "\dashv"] & & \mathbb{E} \arrow[loop above, "\hat{T}"] \arrow[d, "p"] \\
\mathrm{Base} \arrow[ur, "\hat{s}"] \arrow[r, "s"'] & \mathbb{C} \arrow[loop below, "S"] \arrow[rr, "{(F, A)}"'] & & \mathbb{B} \arrow[loop below, "T"]
\end{tikzcd}
$$

strong monad map

DEFINING A LOGICAL RELATION

$$\begin{tikzcd}
 & \hat{C} \arrow[loop above, "\hat{S}"] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & E \arrow[loop above, "\hat{T}"] \arrow[d, "p"] \\
Syn \arrow[ur, "\hat{S}[\![-]\!]"] \arrow[r, "S[\![-]\!]"'] & C \arrow[loop below, "S"] \arrow[r, "{(F,\lambda)}"'] & B \arrow[loop below, "T"]
\end{tikzcd}$$

strong monad map

ie. we have

$$\hat{S}[\![\sigma]\!] = (S[\![\sigma]\!], R_\sigma)$$

so we get

$$R := \{R_\sigma \mid \underbrace{\sigma \in Type}_{\text{including } T\tau?}\}$$

EXAMPLE : GLOBAL STATE

for $P \subseteq \mathbb{C}(1,S)$ any submonoid

$$\begin{tikzcd}
 & \mathrm{Pred} \arrow[d, "p"] \arrow[loop right, "(-)\times P"] \\
\mathbb{C} \arrow[r, "\mathbb{C}(1,-)"'] \arrow[loop below, "(-)\times S"] & \mathrm{Set} \arrow[loop right, "(-)\times \mathbb{C}(1,S)"]
\end{tikzcd}$$

for S a monoid in $\mathbb{C}$

EXAMPLE : GLOBAL STATE

$$\begin{tikzcd}
\hat{\mathbb{C}} \arrow[loop above, "\hat{F}"] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Pred} \arrow[loop right, "(-)\times P"] \arrow[d, "p"] \\
\mathbb{C} \arrow[loop below, "(-)\times S"] \arrow[r, "\mathbb{C}(1,-)"'] & \mathrm{Set} \arrow[loop right, "(-)\times \mathbb{C}(1,S)"]
\end{tikzcd}$$

for $P \leq \mathbb{C}(1,S)$ any submonoid

for S a monoid in $\mathbb{C}$

Example : Global state

$$\begin{tikzcd}
\hat{\mathbb{C}} \arrow[loop above, "\hat{T}"] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Pred} \arrow[loop right, "(-)\times P"] \arrow[d, "p"] \\
\mathbb{C} \arrow[loop below, "(-)\times S"] \arrow[r, "\mathbb{C}(1,-)"'] & \mathrm{Set} \arrow[loop right, "(-)\times \mathbb{C}(1,S)"]
\end{tikzcd}$$

for $P \subseteq \mathbb{C}(1,S)$ any submonoid

for S a monoid in $\mathbb{C}$

$\hat{\mathbb{C}}$ has objects $\overbrace{(X \in \mathbb{C},\ R \subseteq \mathbb{C}(1,X))}^{\text{relational model!}}$

$$\hat{T}(X,R) = (X \times S,\ \hat{T}R)$$

$$\langle x, s\rangle \in \hat{T}R \iff x \in R \text{ and } s \in P \subseteq \mathbb{C}(1,S)$$

EXAMPLE : GLOBAL STATE

$$\begin{tikzcd}
\hat{\mathbb{C}} \arrow[r] \arrow[d] \arrow[loop above, "\hat{T}"] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Pred} \arrow[d, "p"] \arrow[loop right, "(-)\times P"] \\
\mathbb{C} \arrow[r, "\mathbb{C}(1,-)"'] \arrow[loop below, "(-)\times S"] & \mathrm{Set} \arrow[loop right, "(-)\times \mathbb{C}(1,S)"]
\end{tikzcd}$$

for $P \leq \mathbb{C}(1,S)$ any submonoid

for S a monoid in $\mathbb{C}$

$\hat{\mathbb{C}}$ has objects $\underbrace{(x \in \mathbb{C},\ R \subseteq \mathbb{C}(1,x))}$ relational model!

$\hat{T}(X,R) = (X \times S, \hat{T}R)$

$\langle x, s\rangle \in \hat{T}R \Leftrightarrow x \in R$ and $s \in P \leq \mathbb{C}(1,S)$

for a logical relation:

$R_{T\sigma} := \{(x,s) \mid x \in R_\sigma \text{ and } s \in P\}$

EXAMPLE : SIMULATING EFFECTS

à la Katsumata

STRONG MONAD MAP

$$\gamma : L \longrightarrow \mathcal{P}_{fin}$$

$$[x_1, \ldots, x_n] \longmapsto \{x_1, \ldots, x_n\}$$

EXAMPLE : COMPARING EFFECTS

à la Katsumata

$$\gamma : L \longrightarrow \mathcal{P}_{fin}$$

$$[x_1, \ldots, x_n] \longmapsto \{x_1, \ldots, x_n\}$$

$$\begin{array}{c} \mathbf{Set} \times \mathbf{Set} \\ \circlearrowright \\ L \times \mathcal{P}_{fin} \end{array}$$

EXAMPLE : COMPARING EFFECTS

à la Katsumata

$\gamma : L \longrightarrow \mathcal{P}_{fin}$

$[x_1, \dots, x_n] \longmapsto \{x_1, \dots, x_n\}$

$$\begin{tikzcd} & & \mathbf{BinPred} \arrow[loop above, "\hat{T}"] \arrow[d, "p"] \\ \mathbf{Set} \times \mathbf{Set} \arrow[loop below, "L \times \mathcal{P}_{fin}"] \arrow[rr, "(id, \gamma \times id)"'] & & \mathbf{Set} \times \mathbf{Set} \arrow[loop below, "\mathcal{P}_{fin} \times \mathcal{P}_{fin}"] \end{tikzcd}$$

$\underline{obj}$: $(X, Y, R \rightarrowtail X \times Y)$

$\hat{T}(X, Y, R) = (\mathcal{P}_{fin} X, \mathcal{P}_{fin} Y, \mathcal{P}_{fin} R)$

$\mathcal{P}_{fin} R \rightarrowtail \mathcal{P}_{fin} X \times \mathcal{P}_{fin} Y$

Example: Comparing effects

à la Katsumata

$$\gamma : L \longrightarrow \mathcal{P}_{fin}$$
$$[x_1, \ldots, x_n] \longmapsto \{x_1, \ldots, x_n\}$$

$$\begin{tikzcd}
\mathrm{BinPred} \arrow[loop above, "\hat{S}"] \arrow[r] \arrow[d] & \mathrm{BinPred} \arrow[loop above, "\hat{T}"] \arrow[d, "p"] \\
\mathrm{Set} \times \mathrm{Set} \arrow[loop below, "L \times \mathcal{P}_{fin}"] \arrow[r, "(id, \gamma \times id)"'] & \mathrm{Set} \times \mathrm{Set} \arrow[loop below, "\mathcal{P}_{fin} \times \mathcal{P}_{fin}"]
\end{tikzcd}$$

obj: $(X, Y, R \rightarrowtail X \times Y)$

$$\hat{T}(X, Y, R) = (\mathcal{P}_{fin} X, \mathcal{P}_{fin} Y, \mathcal{P}_{fin} R)$$
$$\mathcal{P}_{fin} R \rightarrowtail \mathcal{P}_{fin} X \times \mathcal{P}_{fin} Y$$

$$\hat{S}(X, Y, R) = (LX, \mathcal{P}_{fin} Y, \hat{S}R)$$

$$([x_1, \ldots, x_n], \{y_1, \ldots, y_m\}) \in \hat{S}R$$
$$\iff n = m \text{ and } (x_i, y_i) \in R$$
$$\text{for } i = 1, \ldots, n$$

EXAMPLE : COMPARING EFFECTS

à la Katsumata

$\gamma : L \longrightarrow \mathcal{P}_{fin}$

$[x_1, \ldots, x_n] \longmapsto \{x_1, \ldots, x_n\}$

$$\begin{tikzcd}
\mathrm{BinPred} \arrow[loop above, "\hat{S}"] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{BinPred} \arrow[loop above, "\hat{T}"] \arrow[d, "p"] \\
\mathrm{Set} \times \mathrm{Set} \arrow[loop below, "L \times \mathcal{P}_{fin}"] \arrow[r, "(id, \gamma \times id)"'] & \mathrm{Set} \times \mathrm{Set} \arrow[loop below, "\mathcal{P}_{fin} \times \mathcal{P}_{fin}"]
\end{tikzcd}$$

$\underline{\mathrm{obj}} : (X, Y, R \rightarrowtail X \times Y)$

$\hat{T}(X, Y, R) = (\mathcal{P}_{fin} X, \mathcal{P}_{fin} Y, \mathcal{P}_{fin} R)$

$\mathcal{P}_{fin} R \rightarrowtail \mathcal{P}_{fin} X \times \mathcal{P}_{fin} Y$

$\hat{S}(X, Y, R) = (LX, \mathcal{P}_{fin} Y, \hat{S}R)$

$([x_1, \ldots, x_n], \{y_1, \ldots, y_m\}) \in \hat{S}R$

$\iff n = m$ and $(x_i, y_i) \in R$ for $i = 1, \ldots, n$

LOGICAL RELATION $\mathcal{M}$:

$([x_1, \ldots, x_n], \{y_1, \ldots, y_m\}) \in \mathcal{M}_{T\sigma} \iff m = n$ and each $(x_i, y_i) \in \mathcal{M}_\sigma$

DIFFICULT BIT: defining the target lifting

$$\begin{tikzcd} & 2^{\hat{T}} \\ & \mathbb{E} \arrow[d] \\ \mathbb{C} \arrow[r, "F"'] \arrow[loop below, "S"] & \mathbb{B} \arrow[loop below, "T"] \end{tikzcd}$$

how to get this?

↳ well-developed theory!

eg. Kammar-McDermott,
Katsumata,
Goubault-Larrecq et al.,
...

given by the model

usually a natural target

EXAMPLE: $\top\top$-LIFTING (Katsumata)

for $R \in \mathbf{Set}$ and $\bar{R} \subseteq TR$

$$\begin{tikzcd} \mathbf{Set} \arrow[loop below, "T"] \end{tikzcd}$$

Example: $\top\top$-lifting (Katsumata)

for $R \in \mathrm{Set}$ and $\bar{R} \subseteq TR$

$$\hat{K} := (- \Rightarrow (TR, \bar{R})) \Rightarrow (TR, \bar{R})$$

$$\begin{tikzcd}
 & \mathrm{Pred} \arrow[loop above, "\hat{K}"] \arrow[d, "p"] \\
\mathrm{Set} \arrow[loop below, "T"] \arrow[r, "{(id, \lambda)}"'] & \mathrm{Set} \arrow[loop below, "K"]
\end{tikzcd}$$

$$K := (- \Rightarrow TR) \Rightarrow TR$$

$$\lambda_X : TX \longrightarrow (X \Rightarrow TR) \Rightarrow TR$$

$$t \longmapsto \lambda f : X \to TR.\ \mathrm{let}\ x = t\ \mathrm{in}\ f x$$

EXAMPLE : $\top\top$-LIFTING (Katsumata)

for $R \in \mathbf{Set}$ and $\bar{R} \subseteq TR$

$$\begin{tikzcd}
\mathrm{Pred} \arrow[r] \arrow[d, "p"'] \arrow[loop above, "\hat{T}"] & \mathrm{Pred} \arrow[d, "p"] \arrow[loop above, "\hat{K}"] \\
\mathbf{Set} \arrow[r, "{(id,\, \lambda)}"'] \arrow[loop below, "T"] & \mathbf{Set} \arrow[loop below, "K"]
\end{tikzcd}$$

$\hat{K} := (- \Rightarrow (TR, \bar{R})) \Rightarrow (TR, \bar{R})$

$K := (- \Rightarrow TR) \Rightarrow TR$

$$\lambda_X : TX \longrightarrow (X \Rightarrow TR) \Rightarrow TR$$

$$t \longmapsto \lambda f : X \to TR.\ \text{let } x = t \text{ in } fx$$

$$\hat{T}(X, \bar{X}) = (TX, \hat{T}\bar{X})$$

$t \in \hat{T}\bar{X} \iff \forall f \in (\bar{X} \Rightarrow \bar{R}).$

$\qquad \text{let } x = t \text{ in } fx \in \bar{R}$

"for all nice continuations f, $f \gg= t$ is nice"

cf. Krivine realisability

EXAMPLE : TT-LIFTING (Katsumata)

for $R \in \mathrm{Set}$ and $\bar{R} \subseteq TR$

$$
\begin{tikzcd}
\mathrm{Pred} \arrow[loop above, "\hat{T}"] \arrow[r] \arrow[d, "p"'] & \mathrm{Pred} \arrow[loop above, "\hat{K}"] \arrow[d, "p"] \\
\mathrm{Set} \arrow[loop below, "T"] \arrow[r, "{(id,\, \lambda)}"'] & \mathrm{Set} \arrow[loop below, "K"]
\end{tikzcd}
$$

$\hat{K} := (- \Rightarrow (TR, \bar{R})) \Rightarrow (TR, \bar{R})$

$K := (- \Rightarrow TR) \Rightarrow TR$

$\lambda_X : TX \longrightarrow (X \Rightarrow TR) \Rightarrow TR$

$t \longmapsto \lambda f : X \to TR.\ \mathrm{let}\ x = t\ \mathrm{in}\ fx$

LOGICAL RELATION S:

$t \in S_{T\sigma} \iff \forall f \in (S_\sigma \supset \bar{R}).\ \mathrm{let}\ x = t\ \mathrm{in}\ fx \in \bar{R}$

$\hat{T}(X, \bar{X}) = (TX, \hat{T}\bar{X})$

$t \in \hat{T}\bar{X} \iff \forall f \in (\bar{X} \supset \bar{R}).\ \mathrm{let}\ x = t\ \mathrm{in}\ fx \in \bar{R}$

"for all nice continuations f, $f \gg= t$ is nice"

cf. Krivine realisability

Just as we identified STLC logical relations with model-preserving fibrations, so we can do for CBV models

SUMMING UP

: we have a nice denotational account of CBV logical relations

SUMMING UP

: we have a nice denotational account of CBV logical relations

① denotationally, logical relations is about constructing refined models over existing ones

SUMMING UP

: we have a nice denotational account of CBV logical relations

① denotationally, logical relations is about constructing refined models over existing ones

② such models can be identified as fibrations which strictly preserve the CCC + monad structure – i.e. the model

SUMMING UP

: we have a nice denotational account of CBV logical relations

1. denotationally, logical relations is about constructing refined models over existing ones

2. such models can be identified as fibrations which strictly preserve the CCC + monad structure — ie. the model

3. the general theory gives us a useful framework for finding such fibrations (= relations models)

Summing up

: we have a nice denotational account of CBV logical relations

① denotationally, logical relations is about constructing refined models over existing ones

② such models can be identified as fibrations which strictly preserve the ccc + monad structure — i.e. the model

③ the general theory gives us a useful framework for finding such fibrations (= relations models)

DEFN : a <u>CBV fibration</u> is a fibration that strictly preserves ccc + the monad $\approx$ a logical relation

④ How does this picture extend to CBPV?

Why CBPV?

Subsumes both CBV and CBN

... by giving fine control over when effects happen

FEATURES OF CBPV

- Computations and values separate, but related:

$$\begin{tikzcd} \text{Values} \arrow[r, bend left, "\text{force}"] \arrow[r, phantom, "\perp"] & \text{Computations} \arrow[l, bend left, "\text{thunk}"] \end{tikzcd}$$

no "accidental" reductions - contrast to λ_{ml}

FEATURES OF CBPV

- Computations and values separate, but related:

$$\begin{tikzcd} \text{Values} \arrow[r, bend left, "\text{force}"] \arrow[r, phantom, "\perp"] & \text{Computations} \arrow[l, bend left, "\text{thunk}"] \end{tikzcd}$$

no "accidental" reductions - contrast to λ_{ml}

best of both worlds

- "sums are nice in CBV" + "arrows are nice in CBN"

ie. the sum type consists of things like $inj_i\, V$ for V a value

in CBN, effects commute with λ so the η-law holds

FEATURES OF CBPV

- Computations and values separate, but related:

$$\begin{tikzcd} \text{Values} \arrow[r, bend left, "\text{force}"] \arrow[r, phantom, "\perp"] & \text{Computations} \arrow[l, bend left, "\text{thunk}"] \end{tikzcd}$$

no "accidental" reductions - contrast to λ_{ml}

best of both worlds

- "sums are nice in CBV" + "arrows are nice in CBN"

  i.e. the sum type consists of things like $\mathrm{inj}_i\ V$ for V a value

  in CBN, effects commute with λ so the η-law holds

- let-binding into any computation type, not just free ones

  ↳ important for getting good embeddings of e.g. sum types from λ_{ml}

Some syntax

$$\begin{tikzcd} \text{Values} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \underline{\text{Computations}} \arrow[l, bend left, "U"] \end{tikzcd}$$

"computations do, values are"

Some syntax

$$\begin{tikzcd} \text{Values} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \underline{\text{Computations}} \arrow[l, bend left, "U"] \end{tikzcd}$$

"computations <u>do</u>, values <u>are</u>"

contexts are values

$$\frac{\Gamma \vdash^v V : A}{\Gamma \vdash^c \text{return } V : FA}$$

values are trivial computations

$$\frac{\Gamma \vdash^c M : \underline{B}}{\Gamma \vdash^v \text{thunk } M : U\underline{B}}$$

prevent a computation running

$$\frac{\Gamma \vdash^v V : U\underline{B}}{\Gamma \vdash^c \text{force } V : \underline{B}}$$

allow the computation to run

Some syntax

$$\begin{tikzcd} \text{Values} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \underline{\text{Computations}} \arrow[l, bend left, "U"] \end{tikzcd}$$

"computations do, values are"

contexts are values

$$\frac{\Gamma \vdash^v V : A}{\Gamma \vdash^c \text{return } V : FA} \qquad \frac{\Gamma \vdash^c M : \underline{B}}{\Gamma \vdash^v \text{thunk } M : U\underline{B}} \qquad \frac{\Gamma \vdash^v V : U\underline{B}}{\Gamma \vdash^c \text{force } V : \underline{B}}$$

values are trivial computations

prevent a computation running

allow the computation to run

CBV function type becomes

$$\ulcorner A \to_{cbv} B \urcorner = U(\underbrace{A \to FB}_{\text{function types are computations}})$$

$$\underbrace{\ulcorner \lambda x.\, M \urcorner}_{\text{value in CBV}} = \underbrace{\text{thunk}}_{\text{makes it a value}}\ \lambda x.\, \ulcorner M \urcorner$$

Some syntax

$$\begin{tikzcd} \text{Values} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \underline{\text{Computations}} \arrow[l, bend left, "U"] \end{tikzcd}$$

"computations $\underline{\text{do}}$, values $\underline{\text{are}}$"

contexts are values

$$\frac{\Gamma \vdash^v V : A}{\Gamma \vdash^c \text{return } V : FA} \qquad \frac{\Gamma \vdash^c M : \underline{B}}{\Gamma \vdash^v \text{thunk } M : U\underline{B}} \qquad \frac{\Gamma \vdash^v V : U\underline{B}}{\Gamma \vdash^c \text{force } V : \underline{B}}$$

values are trivial computations

prevent a computation running

allow the computation to run

$\underline{\text{CBV}}$ function type becomes

$$\ulcorner A \to_{CBV} B \urcorner = U(\underbrace{A \to FB}_{\text{function types are computations}})$$

$$\underbrace{\ulcorner \lambda x.\, M \urcorner}_{\text{value : CBV}} = \underbrace{\text{thunk}}_{\text{makes it a value}}\ \lambda x.\, \ulcorner M \urcorner$$

$\underline{\text{CBN}}$ function type becomes

$$\ulcorner A \to_{CBN} B \urcorner = \underbrace{(UA)}_{} \to B$$

in CBN variables are bound to unevaluated terms, so we regard these as thunks

Some syntax

$$\begin{tikzcd} \text{Values} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \underline{\text{Computations}} \arrow[l, bend left, "U"] \end{tikzcd}$$

"computations do, values are"

contexts are values

$$\frac{\Gamma \vdash^v V : A}{\Gamma \vdash^c \text{return } V : FA}$$

values are trivial computations

$$\frac{\Gamma \vdash^c M : \underline{B}}{\Gamma \vdash^v \text{thunk } M : U\underline{B}}$$

prevent a computation running

$$\frac{\Gamma \vdash^v V : U\underline{B}}{\Gamma \vdash^c \text{force } V : \underline{B}}$$

allow the computation to run

the monad is still there

$$\cfrac{\cfrac{\Gamma \vdash^v V : A}{\Gamma \vdash^c \text{return } V : FA}}{\Gamma \vdash^v \text{thunk}(\text{return } V) : U(FA)}$$

What we want: a denotational account of log. rel. for CBPV

What we want: a denotational account of log. rel. for CBPV

1) CBPV fibration = fibration s.t. model structure preserved

2) universal construction of relations models

$$\begin{tikzcd}
\text{(relational model)} \arrow[r] \arrow[d] \arrow[dr, phantom, very near start, "\lrcorner"] & \text{(relations)} \arrow[d] \\
\text{(starting model)} \arrow[r] & \text{(nice model)}
\end{tikzcd}$$

3) recover sensible syntactic def^n

$\{R_A \mid A \in \mathrm{ValType}\}$

$\{R_B \mid B \in \mathrm{CompType}\}$

Semantics of CBPV [Levy]

Need to interpret

$\Gamma \vdash^v V : A$ values

$\Gamma \vdash^c M : \underline{B}$ computations

$\Gamma \mid \underline{B} \vdash^k K : \underline{C}$ stacks / contexts

cf. $C[M]$ for a λ-calc term M and context $C[-]$

Semantics of CBPV [Levy]

Need to interpret

need homs depending on $\Gamma \in \mathrm{Val}$ and $\underline{B} \in \mathrm{Comp}$

need homs depending on $\Gamma \in \mathrm{Val}$ and $\underline{B}, \underline{C} \in \mathrm{Comp}$

$\Gamma \vdash^v V : A$ — values

$\overbrace{\Gamma \vdash^c M : \underline{B}}$ — computations

$\overbrace{\Gamma \mid \underline{B} \vdash^k K : \underline{C}}$ — stacks / contexts

cf. $C[M]$ for a λ-calc term M and context $C[-]$

Semantics of CBPV [Levy]

Need to interpret

need homs depending on $\Gamma \in \mathrm{Val}$ and $\underline{B} \in \mathrm{Comp}$

need homs depending on $\Gamma \in \mathrm{Val}$ and $\underline{B}, \underline{C} \in \mathrm{Comp}$

$\Gamma \vdash^v V : A$

values

in a cartesian category $\mathbb{V}$, ie. in $\mathbb{V}([\![\Gamma]\!], [\![A]\!])$

$\Gamma \vdash^c M : \underline{B}$

computations

in $\mathbb{V}([\![\Gamma]\!], U[\![\underline{B}]\!])$

$\cong \mathcal{C}_{[\![\Gamma]\!]}(F1, [\![\underline{B}]\!])$

$\Gamma \mid \underline{B} \vdash^k K : \underline{C}$

stacks / contexts

cf. $C[M]$ for a λ-calc term M and context $C[-]$

in a locally $\mathbb{V}$-indexed category $\mathcal{C}$, ie. in $\mathcal{C}_{[\![\Gamma]\!]}([\![\underline{B}]\!], [\![\underline{C}]\!])$

Semantics of CBPV [Levy]

<u>Defn</u>: let $\mathbb{V}$ be cartesian. A <u>locally $\mathbb{V}$-indexed</u> category $\mathcal{C}$ has

- objects $A, B, \ldots$
- for every $\Gamma \in \mathbb{V}$, $A, B \in \mathcal{C}$ a hom set $\mathcal{C}_\Gamma(A, B)$ of arrows $f : A \xrightarrow[\Gamma]{} B$ <u>over Γ</u>

...

Semantics of CBPV [Levy]

<u>DEFN</u>: let $\mathbb{V}$ be cartesian. A <u>locally $\mathbb{V}$-indexed</u> category $\mathcal{C}$ has

- objects $A, B, \ldots$
- for every $\Gamma \in \mathbb{V}$, $A, B \in \mathcal{C}$ a hom set $\mathcal{C}_\Gamma(A, B)$ of arrows $f : A \xrightarrow[\Gamma]{} B$ <u>over Γ</u>

st. each $\mathcal{C}_\Gamma$ B a category with the same objects, compatible with maps in $\mathbb{V}$ //

Semantics of CBPV [Levy]

a category enriched in $[\mathbb{V}^{op}, \mathrm{Set}]$ //

Defn: let $\mathbb{V}$ be cartesian. A locally $\mathbb{V}$-indexed category $\mathscr{C}$ has

- objects $A, B, \ldots$
- for every $\Gamma \in \mathbb{V}$, $A, B \in \mathscr{C}$ a hom set $\mathscr{C}_\Gamma(A, B)$ of arrows $f : A \xrightarrow[\Gamma]{} B$ over Γ

st. each $\mathscr{C}_\Gamma$ is a category with the same objects, compatible with maps in $\mathbb{V}$ //

EXAMPLES for $\mathbb{V}$ cartesian

- self $\mathbb{V}$ has objects as in $\mathbb{V}$ and

$$(\mathrm{self}\ \mathbb{V})_\Gamma(A, B) := \mathbb{V}(\Gamma \times A, B)$$

EXAMPLES for $\mathbb{V}$ cartesian

- self $\mathbb{V}$ has objects as in $\mathbb{V}$ and

$$(\mathrm{self}\ \mathbb{V})_\Gamma (A, B) := \mathbb{V}(\Gamma \times A, B)$$

- if T is a strong monad on $\mathbb{V}$, get $\mathcal{E}(T)$ with objects T-algebras and maps $f : A \to_\Gamma B$ maps $f : \Gamma \times A \longrightarrow B$ in $\mathbb{V}$ that are right-linear

SEMANTICS

$$\left(\begin{array}{c}\text{values trivially} \\ \text{indexed over themselves}\end{array}\right) \underset{\longleftarrow}{\overset{\longrightarrow}{\perp}} \left(\begin{array}{c}\text{computations indexed} \\ \text{over values}\end{array}\right)$$

SEMANTICS

(values trivially indexed over themselves) $\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}$ (computations indexed over values)

DEFN: A <u>CBPV model</u> consists of

- a cartesian category $\mathbb{V}$ for the values

SEMANTICS

$$\left(\begin{array}{l}\text{values trivially}\\ \text{indexed over themselves}\end{array}\right) \underset{\longleftarrow}{\overset{\longrightarrow}{\perp}} \left(\begin{array}{l}\text{computations indexed}\\ \text{over values}\end{array}\right)$$

DEF$^{\underline{N}}$: A <u>CBPV model</u> consists of

- a cartesian category $\mathbb{V}$ for the values
- a locally $\mathbb{V}$-indexed category $\mathcal{C}$ for stacks / computations

SEMANTICS

$$\left(\begin{array}{l}\text{values trivially}\\ \text{indexed over themselves}\end{array}\right) \;\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}\; \left(\begin{array}{l}\text{computations indexed}\\ \text{over values}\end{array}\right)$$

DEF$^{\underline{N}}$: A <u>CBPV model</u> consists of

- a cartesian category $\mathbb{V}$ for the values
- a locally $\mathbb{V}$-indexed category $\mathscr{C}$ for stacks / computations
- a locally $\mathbb{V}$-indexed adjunction i.e. an adjunction in $[\mathbb{V}, \mathbf{Set}]$-$\mathbf{Cat}$

$$\text{self } \mathbb{V} \;\underset{U}{\overset{F}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}}\; \mathscr{C} \qquad \Bigg\} \qquad \mathbb{V}(\Gamma \times X, U\underline{B}) \cong \mathscr{C}_\Gamma(FX, \underline{B})$$

SEMANTICS

(values trivially indexed over themselves) $\rightleftarrows$ $\perp$ (computations indexed over values)

DEF$^{\underline{N}}$: A <u>CBPV model</u> consists of

- a cartesian category $\mathbb{V}$ for the values
- a locally $\mathbb{V}$-indexed category $\mathcal{C}$ for stacks / computations
- a locally $\mathbb{V}$-indexed adjunction i.e. an adjunction in $[\mathbb{V}, \mathrm{Set}]$-Cat

$$\text{self}\ \mathbb{V} \underset{U}{\overset{F}{\rightleftarrows}} \mathcal{C} \quad (\perp)$$

$$\mathbb{V}(\Gamma \times X, UB) \cong \mathcal{C}_\Gamma(FX, \underline{B})$$

+ structure for $\to$, $+$, $\times$ etc.

Semantics

(values trivially indexed over themselves) $\rightleftarrows$ (computations indexed over values)

$\perp$

Value types

$[\![A]\!] \in \mathbb{V}$

Computation types

$[\![B]\!] \in \mathcal{C}$

locally-indexed functor

$$\text{self} \quad \begin{tikzcd} \mathbb{V} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{C} \arrow[l, bend left, "U"] \end{tikzcd}$$

locally-indexed functor

unit + counit give rise to thunk / force

Semantics of CBPV [Levy]

Need to interpret

need homs depending on $\Gamma \in$ Val and $\underline{B} \in$ Comp

need homs depending on $\Gamma \in$ Val and $\underline{B}, \underline{C} \in$ Comp

$\Gamma \vdash^{v} V : A$	$\Gamma \vdash^{c} M : \underline{B}$	$\Gamma \mid \underline{B} \vdash^{k} K : \underline{C}$
values	computations	stacks / contexts

cf. $C[M]$ for a λ-cal term M and context $C[-]$

Semantics of CBPV [Levy]

Need to interpret

need homs depending on $\Gamma \in \mathrm{Val}$ and $\underline{B} \in \mathrm{Comp}$

need homs depending on $\Gamma \in \mathrm{Val}$ and $\underline{B}, \underline{C} \in \mathrm{Comp}$

$\Gamma \vdash^v V : A$

values

$\Gamma \vdash^c M : \underline{B}$

computations

$\Gamma \mid \underline{B} \vdash^k K : \underline{C}$

stacks / contexts

cf. $C[M]$ for a λ-calc term M and context $C[-]$

in a cartesian category $\mathbb{V}$, ie. in $\mathbb{V}(⟦\Gamma⟧, ⟦A⟧)$

in a locally $\mathbb{V}$-indexed category $\mathcal{C}$, ie. in $\mathcal{C}_{⟦\Gamma⟧}(⟦\underline{B}⟧, ⟦\underline{C}⟧)$

Semantics of CBPV [Levy]

Need to interpret

need homs depending on $\Gamma \in Val$ and $\underline{B} \in Comp$

need homs depending on $\Gamma \in Val$ and $\underline{B}, \underline{C} \in Comp$

$\Gamma \vdash^v V : A$

values

in a cartesian category $\mathbb{V}$, ie. in $\mathbb{V}([\![\Gamma]\!], [\![A]\!])$

$\Gamma \vdash^c M : \underline{B}$

computations

in $\mathbb{V}([\![\Gamma]\!], U[\![\underline{B}]\!])$

$\cong \mathcal{C}_{[\![\Gamma]\!]}(F1, [\![\underline{B}]\!])$

$\Gamma \mid \underline{B} \vdash^k K : \underline{C}$

stacks / contexts

cf. $C[M]$ for a λ-term M and context $C[-]$

in a locally $\mathbb{V}$-indexed category $\mathcal{C}$, ie. in $\mathcal{C}_{[\![\Gamma]\!]}([\![\underline{B}]\!], [\![\underline{C}]\!])$

Many more in the CBPV book!

EXAMPLES:

1) ALGEBRA MODELS

for T a strong monad on $\mathbb{V}$

$$\begin{tikzcd} \mathrm{Self}\ \mathbb{V} \arrow[r, bend left, "\text{free}"] \arrow[r, phantom, "\perp"] & \mathcal{E}(T) \arrow[l, bend left, "\text{forget}"] \end{tikzcd}$$

Many more in the CBPV book!

EXAMPLES:

1) **ALGEBRA MODELS**

for T a strong monad on $\mathbb{V}$

$$\begin{tikzcd} \mathrm{self}\ \mathbb{V} \arrow[r, bend left, "\text{free}"] \arrow[r, phantom, "\perp"] & \mathcal{E}(T) \arrow[l, bend left, "\text{forget}"] \end{tikzcd}$$

2) **STATE MODELS**

for a ccc $\mathbb{C}$ and $S \in \mathbb{C}$

$$\begin{tikzcd} \mathrm{self}\ \mathbb{C} \arrow[r, bend left, "S \Rightarrow (-)"] \arrow[r, phantom, "\perp"] & \mathrm{self}\ \mathbb{C} \arrow[l, bend left, "(-) \times S"] \end{tikzcd}$$

Many more in the CBPV book!

EXAMPLES:

1) ALGEBRA MODELS

for T a strong monad on $\mathbb{V}$

$$\begin{tikzcd} \mathrm{self}\ \mathbb{V} \arrow[r, bend left, "\mathrm{free}"] \arrow[r, phantom, "\perp"] & \mathcal{E}(T) \arrow[l, bend left, "\mathrm{forget}"] \end{tikzcd}$$

2) STATE MODELS

for a ccc $\mathbb{C}$ and $S \in \mathbb{C}$

$$\begin{tikzcd} \mathrm{self}\ \mathbb{C} \arrow[r, bend left, "S \Rightarrow (-)"] \arrow[r, phantom, "\perp"] & \mathrm{self}\ \mathbb{C} \arrow[l, bend left, "(-) \times S"] \end{tikzcd}$$

3) CONTINUATION MODELS

for a ccc $\mathbb{C}$ and $R \in \mathbb{C}$

$$\begin{tikzcd} \mathrm{self}\ \mathbb{C} \arrow[r, bend left, "- \Rightarrow R"] \arrow[r, phantom, "\perp"] & \mathrm{self}\ \mathbb{C}^{op} \arrow[l, bend left, "- \Rightarrow R"] \end{tikzcd}$$

What we want: a denotational account of log. rel. for CBPV

1) CBPV fibration = fibration s.t. model structure preserved

2) universal construction of relations models

$$\begin{tikzcd}
(\text{relational model}) \arrow[r] \arrow[d] \arrow[dr, phantom, very near start, "\lrcorner"] & (\text{relations}) \arrow[d] \\
(\text{starting model}) \arrow[r] & (\text{nice model})
\end{tikzcd}$$

3) recover sensible syntactic defn

$\{R_A \mid A \in \mathrm{ValType}\}$

$\{R_B \mid B \in \mathrm{CompType}\}$

Qn: what is a CBPV fibration?

Prior work in the algebras case, see eg. Kammar, McDermott.

Qn: what is a CBPV fibration?

Prior work in the algebras case, see eg. Kammar, McDermott.

It can't be a fibration that preserves the structure!

... at least, not immediately. Why not?

Qn: what is a CBPV fibration?

Prior work in the algebras case, see eg. Kammar, McDermott.

It can't be a fibration that preserves the structure!

... at least, not immediately. Why not?

What is a fibration of locally indexed categories?

<u>Qn</u> : what is a CBPV fibration?

<u>A recipe</u> :

Qn: what is a CBPV fibration?

A recipe: 1) Define a 2-category of locally-indexed categories

Qn: what is a CBPV fibration?

A recipe: 1) Define a 2-category of locally-indexed categories

2) Use this to define locally-indexed fibrations

as fibrations internal to this 2-category

Qn: what is a CBPV fibration?

A recipe: 1) Define a 2-category of locally-indexed categories

2) Use this to define locally-indexed fibrations

3) A CBPV fibration is a locally-indexed fibration that preserves the structure

CBPV fibrations

$$\mathrm{self}\ \mathbb{C} \xrightarrow[F]{} \mathcal{C} \xrightarrow[u]{} \mathrm{self}\ \mathbb{C}$$

CBPV fibrations

$$\mathrm{self}\ \hat{\mathbb{C}} \xrightarrow{\hat{F}} \hat{\mathscr{C}} \xrightarrow{\hat{u}} \mathrm{self}\ \hat{\mathbb{C}}$$

$$\mathrm{self}\ \mathbb{C} \xrightarrow[F]{} \mathscr{C} \xrightarrow[u]{} \mathrm{self}\ \mathbb{C}$$

CBPV fibrations

$$
\begin{tikzcd}
\mathrm{self}\ \hat{\mathbb{C}} \arrow[r, "\hat{F}"] \arrow[d, "\mathrm{self}\ p"'] & \hat{\mathcal{C}} \arrow[r, "\hat{U}"] \arrow[d, "p"] & \mathrm{self}\ \hat{\mathbb{C}} \arrow[d, "\mathrm{self}\ p"] \\
\mathrm{self}\ \mathbb{C} \arrow[r, "F"'] & \mathcal{C} \arrow[r, "U"'] & \mathrm{self}\ \mathbb{C}
\end{tikzcd}
$$

fibration

locally indexed fibration

in the 2-category of locally-indexed categories

CBPV fibrations

$$
\begin{tikzcd}
\mathrm{self}\ \hat{\mathbb{C}} \arrow[r, "\hat{F}"] \arrow[d, "\mathrm{self}\ p"'] & \hat{\mathcal{C}} \arrow[r, "\hat{U}"] \arrow[d, "p"] & \mathrm{self}\ \hat{\mathbb{C}} \arrow[d, "\mathrm{self}\ p"] \\
\mathrm{self}\ \mathbb{C} \arrow[r, "F"'] & \mathcal{C} \arrow[r, "U"'] & \mathrm{self}\ \mathbb{C}
\end{tikzcd}
$$

fibration

locally indexed

... st the adjunction structure is presened

from the general theory, we get:

= some nice 2-category theory

from the general theory, we get:

1) nice closure properties

from the general theory, we get:

1) nice closure properties

2) universal constructions of relational models:

$$
\begin{tikzcd}
\hat{\mathcal{M}} \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathcal{E} \arrow[d, "\text{CBPV fibration}"] \\
\underset{\text{CBPV model}}{\mathcal{M}} \arrow[r, "\text{prod-preserving map}"'] & \mathcal{B}
\end{tikzcd}
$$

from the general theory, we get:

1) nice closure properties

2) universal constructions of relational models:

$$\begin{tikzcd} \hat{\mathcal{M}} \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathcal{E} \arrow[d, "\text{CBPV fibration}"] \\ \text{CBPV model} \quad \mathcal{M} \arrow[r, "\text{prod-preserving map}"'] & \mathcal{B} \end{tikzcd}$$

3) locally-indexed versions of key fibrations for new-from-old (eg. subobject, codomain)

+ recovers the syntactic def^{n} of McDermott

from the general theory, we get:

1) nice closure properties

2) universal constructions of relational models:

$$\begin{tikzcd}
\hat{\mathcal{M}} \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathcal{E} \arrow[d, "\text{CBPV fibration}"] \\
\underset{\text{CBPV model}}{\mathcal{M}} \arrow[r, "\text{prod-preserving map}"'] & \mathcal{B}
\end{tikzcd}$$

* it really is the same if you do enough abstract nonsense

3) locally-indexed versions of key fibrations for new-from-old (eg. subobject, codomain)

\+ recovers the syntactic def$^{\underline{n}}$ of McDermott

~ so essentially the theory we had for STLC / CBV*

EXAMPLES.

1) **algebra models** : coincides with monad lifting

Examples.

1) **algebra models**: coincides with monad lifting

2) **state / continuation models as expected**:

$$
\begin{tikzcd}
\mathrm{Self}\,\hat{\mathbb{C}} \arrow[r, bend left, "(S, \bar{S}) \Rightarrow -"] \arrow[r, phantom, "\perp"] \arrow[d] & \mathrm{Self}\,\hat{\mathbb{C}} \arrow[l, bend left, "(-) \times (S, \bar{S})"] \arrow[d] \arrow[rr] & & \mathrm{Self}\,\mathrm{Pred} \arrow[r, bend left] \arrow[r, phantom, "\perp"] \arrow[d] & \mathrm{Self}\,\mathrm{Pred} \arrow[l, bend left] \arrow[d] \\
\mathrm{Self}\,\mathbb{C} \arrow[r, bend left, "S \Rightarrow -"] \arrow[r, phantom, "\perp"] & \mathrm{Self}\,\mathbb{C} \arrow[l, bend left, "S \times -"] \arrow[rr] & & \mathrm{Self}\,\mathrm{Set} \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \mathrm{Self}\,\mathrm{Set} \arrow[l, bend left]
\end{tikzcd}
$$

EXAMPLES.

1) **algebra models** : coincides with monad lifting

2) **state / continuation models as expected** :

$$
\begin{tikzcd}
\mathrm{Self}\,\hat{\mathbb{C}} \arrow[r, bend left, "(s{,}\bar{s}) \Rightarrow -"] \arrow[d] \arrow[r, phantom, "\perp"] & \mathrm{Self}\,\hat{\mathbb{C}} \arrow[l, bend left, "(-)\times(s{,}\bar{s})"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Self}\,\mathrm{Pred} \arrow[r, bend left] \arrow[d] \arrow[r, phantom, "\perp"] & \mathrm{Self}\,\mathrm{Pred} \arrow[l, bend left] \arrow[d] \\
\mathrm{Self}\,\mathbb{C} \arrow[r, bend left, "S \Rightarrow -"] \arrow[r, phantom, "\perp"] & \mathrm{Self}\,\mathbb{C} \arrow[l, bend left, "S \times -"] & \mathrm{Self}\,\mathrm{Set} \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \mathrm{Self}\,\mathrm{Set} \arrow[l, bend left]
\end{tikzcd}
$$

$\rightsquigarrow$ **get logical relations as families of relations**

$\{R_A \mid A \in \mathrm{ValType}\}$, $\{R_{\underline{B}} \mid \underline{B} \in \mathrm{CompType}\}$

EFFECT SIMULATION

for algebra models for $\underbrace{L}_{\text{algebras = monoids}}$ and $\underbrace{S_{fin}}_{\text{algebras = sup-semilattices}}$ on Set

EFFECT SIMULATION

for algebra models for $\underbrace{L}_{\substack{\text{algebras} \\ = \text{monoids}}}$ and $\underbrace{S_{fin}}_{\substack{\text{algebras} \\ = \text{sup-semilattices}}}$ on Set

obj: $(X, Y, R \subseteq X \times Y)$ but X, Y are monoids and $R \rightarrowtail X \times Y$

↳ takes the effect into account!

$$
\begin{tikzcd}
 & & \mathbf{BinPred} \arrow[r, shift left] \arrow[r, phantom, "\perp"] \arrow[d] & \mathbf{BinPredMon} \arrow[l, shift left] \\
\mathbf{Set} \times \mathbf{Set} \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathbf{SLat} \times \mathbf{Mon} \arrow[l, shift left] \arrow[r, "H"'] & \mathbf{Set} \times \mathbf{Set} \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathbf{Mon} \times \mathbf{Mon} \arrow[l, shift left]
\end{tikzcd}
$$

EFFECT SIMULATION

for algebra models for L and $\mathcal{P}_{fin}$ on Set

L: algebras = monoids

$\mathcal{P}_{fin}$: algebras = sup-semilattices

obj: $(\underset{\text{sup semilattice}}{X}, \overset{\text{monoid}}{M}, \underset{\text{submonoid}}{R \subseteq HX \times M})$

$$\begin{array}{ccc}
\mathrm{BinPred} \underset{\hat{u}}{\overset{\hat{F}}{\underset{\perp}{\rightleftarrows}}} \hat{\mathcal{C}} & \longrightarrow & \mathrm{BinPred} \underset{\perp}{\rightleftarrows} \mathrm{BinPredMon} \\
\downarrow & & \downarrow \\
\mathrm{Set} \times \mathrm{Set} \underset{\perp}{\rightleftarrows} \mathrm{SLat} \times \mathrm{Mon} & \xrightarrow[H]{} & \mathrm{Set} \times \mathrm{Set} \underset{\perp}{\rightleftarrows} \mathrm{Mon} \times \mathrm{Mon}
\end{array}$$

obj: $(X, Y, R \subseteq X \times Y)$ but X, Y are monoids and $R \rightarrowtail X \times Y$

↳ takes the effect into account!

EFFECT SIMULATION

for algebra models for $\mathcal{L}$ and $\mathcal{P}_{fin}$ on Set

$\underbrace{\mathcal{L}}_{\text{algebras = monoids}}$ $\underbrace{\mathcal{P}_{fin}}_{\text{algebras = sup-semilattices}}$

obj : $(\underset{\text{sup semilattice}}{X}, \overset{\text{monoid}}{M}, \underset{\text{submonoid}}{R \subseteq HX \times M})$

obj : $(X, Y, R \subseteq X \times Y)$ but X, Y are monoids and $R \rightarrowtail X \times Y$

↳ takes the effect into account!

$$\begin{tikzcd}
\mathrm{BinPred} \arrow[r, shift left, "\hat{F}"] \arrow[d] & \hat{\mathcal{C}} \arrow[l, shift left, "\hat{U}"] \arrow[r, phantom, "\longrightarrow"] & \mathrm{BinPred} \arrow[r, shift left] & \mathrm{BinPredMon} \arrow[l, shift left] \arrow[d] \\
\mathrm{Set} \times \mathrm{Set} \arrow[r, shift left] & \mathrm{SLat} \times \mathrm{Mon} \arrow[l, shift left] \arrow[r, "H"'] & \mathrm{Set} \times \mathrm{Set} \arrow[r, shift left] & \mathrm{Mon} \times \mathrm{Mon} \arrow[l, shift left]
\end{tikzcd}$$

$\hat{F}(A, B, R) = (\mathcal{P}_{fin} A, \mathcal{L} B, \hat{F} R)$ where $(p, \ell) \in \hat{F} R$ iff a "bisimulation" property holds

WE NOW HAVE

- framework for building lots of CBPV models
- denotational view on CBPV logical relations
- mathematical account encompassing both
  CBV and CBPV
  monad models   adjunction models

Summing up:

- logical relations, denotationally $\simeq$ fibrations that preserve the model

  gives an elegant framework for relational models; justifies semantically the "logical relations conditions"

Summing up:

- logical relations, denotationally $\simeq$ fibrations that preserve the model

  gives an elegant framework for relational models; justifies semantically the "logical relations conditions"

- for CBPV this is harder, because the models are not plain categories

Summing up:

- logical relations, denotationally $\simeq$ fibrations that preserve the model

  gives an elegant framework for relational models; justifies semantically the "logical relations conditions"

- for CBPU this is harder, because the models are not plain categories

- ... but by using some 2-category theory we can get back to a framework as nice as that for STLC / CBV.