

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

Juv. 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 $\{R_\sigma \subseteq [D]^n \mid \sigma \in \text{Type}\}$
such that

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

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

is a term formation rule, then:

$$\begin{array}{l} ([t_i]_D, \dots, [t_i]_D) \in R_{\tau_i} \\ \text{for all } i \end{array} \quad \Rightarrow \quad ([op(t_1, t_2)]_D, \dots, [op(t_1, t_2)]_D) \in R_\tau$$

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

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

is a term formation rule, then:

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

BASIC LEMMA: if $\vdash t : \sigma$ then $([t]_D, \dots, [t]_D) \in R_\sigma$.
(so long as this is true at base types)

How do we use logical relations?

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

Is x the interpretation of some term?

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

Is x the interpretation of some term?

STS : there is some logical relation R and type σ
st. $(x_1, \dots, x) \notin R_\sigma$.

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

Is x the interpretation of some term?

STS: there is some logical relation R and type σ
st. $(x_1, \dots, x) \notin R_\sigma$.

eg// [Plotkin, Sieber]

- definability of elements in Scott's D_∞ model
- parallel-or is not definable in PCF

↳ sparked a huge literature on full abstraction!

[Milner '74]

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

$$\llbracket t \rrbracket^p = \{x_1, \dots, x_n\} \iff \llbracket t \rrbracket^{\text{list}} = \left(\begin{array}{c} \text{some permutation of} \\ [x_1, \dots, x_n] \end{array} \right) ?$$

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

$$\llbracket t \rrbracket^P = \{x_1, \dots, x_n\} \iff \llbracket t \rrbracket^{\text{list}} = \left(\begin{array}{c} \text{some permutation of} \\ \{x_1, \dots, x_n\} \end{array} \right) ?$$

STRATEGY : define a logical relation R on $M \times N$ saying they have the same behaviour at base types. Then the interpretation of $t : \sigma$ is a pair $(\llbracket t \rrbracket^M, \llbracket t \rrbracket^N) \in R_\sigma$ 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

$$\llbracket t \rrbracket^P = \{x_1, \dots, x_n\} \iff \llbracket t \rrbracket^{List} = \left(\begin{array}{l} \text{some permutation of} \\ \{x_1, \dots, x_n\} \end{array} \right) ?$$

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

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

① What is a logical relation?

(b) for STLC

STLC = Simply-typed λ -calculus

types $\tau ::= \beta \in \text{Base} \mid 1 \mid \tau_1 \times \tau_2 \mid \sigma \rightarrow \tau$

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

eqns $\pi_i \langle t_1, t_2 \rangle \stackrel{\beta}{=} t_i$, $(\lambda x. t) u \stackrel{\beta}{=} t[x/u]$

$t \stackrel{\eta}{=} \langle \pi_1 t, \pi_2 t \rangle$, $t \stackrel{\eta}{=} \lambda x. t^x x$

$() \stackrel{\eta}{=} t$ (for $t : 1$)

STLC = Simply-typed λ -calculus

types $\tau ::= \beta \in \text{Base} \mid 1 \mid \tau_1 \times \tau_2 \mid \sigma \rightarrow \tau$

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

eqns $\pi_i \langle t_1, t_2 \rangle \stackrel{\beta}{=} t_i$, $(\lambda x. t) u \stackrel{\beta}{=} t[x/u]$

$t \stackrel{\eta}{=} \langle \pi_1 t, \pi_2 t \rangle$, $t \stackrel{\eta}{=} \lambda x. t^x x$

$() \stackrel{\eta}{=} 1$ (for $t : 1$)

SEMANTIC MODEL :

c.c.c $\mathcal{C} + s : \text{Base} \longrightarrow \mathcal{C}$ interpreting base types

... for a set-like model:

DEFⁿ: an ^{unary} (STLC) logical relation R is a family of relations $\{R_\sigma \subseteq [\sigma] \mid \sigma \in \text{Type}\}$ st.

... for a set-like model:

DEFⁿ: an ^{unary} (STLC) logical relation R is a family of relations $\{R_\sigma \subseteq [\sigma] \mid \sigma \in \text{Type}\}$ st.

$$\textcircled{1} \quad R_1 = \{*\}$$

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

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

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

... for a set-like model:

DEFⁿ: an ^{unary} (STLC) logical relation R is a family of relations $\{R_\sigma \subseteq [\sigma] \mid \sigma \in \text{Type}\}$ st.

$$\textcircled{1} \quad R_1 = \{*\}$$

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

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

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

$$\textcircled{3} \quad f \in R_{\sigma \rightarrow \tau} \subseteq [\sigma] \rightarrow [\tau]$$

$$\iff \forall x \in R_\sigma \subseteq [\sigma]. \quad f x \in R_\tau \subseteq [\tau] \quad //$$

... for a CCC \mathcal{C} :

DEFⁿ: an ^{unary} (STLC) logical relation R is a family of relations $\{R_\sigma \subseteq \mathcal{C}(1, \llbracket \sigma \rrbracket) \mid \sigma \in \text{Type}\}$ st.

① $R_1 = \{*\}$

② $\langle x_1, x_2 \rangle \in R_{\sigma_1 \times \sigma_2} \subseteq \mathcal{C}(1, \llbracket \sigma_1 \rrbracket \times \llbracket \sigma_2 \rrbracket)$

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

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

③ $f \in R_{\sigma \rightarrow \tau} \subseteq \mathcal{C}(1, \llbracket \sigma \rrbracket \Rightarrow \llbracket \tau \rrbracket)$

$\iff \forall x \in R_\sigma \subseteq \mathcal{C}(1, \llbracket \sigma \rrbracket).$

$1 \xrightarrow{\langle f, * \rangle} (\llbracket \sigma \rrbracket \Rightarrow \llbracket \tau \rrbracket) \times \llbracket \sigma \rrbracket \xrightarrow{\text{eval}} \llbracket \tau \rrbracket \in R_\tau$

DEFⁿ: an ^{unary} (STLC) logical relation R is a family of relations $\{R_\sigma \subseteq \mathcal{C}(1, [\sigma]) \mid \sigma \in \text{Type}\}$ st.

① $R_1 = \{*\}$

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

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

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

③ $f \in R_{\sigma \rightarrow \tau} \subseteq \mathcal{C}(1, [\sigma] \rightarrow [\tau])$

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

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

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 SLC et al, which solve slightly different problems. (eg. adding sums or other types)

② How do we understand
logical relations denotationally?

BACK TO STLC IN SET :

given $s: \text{Base} \rightarrow \text{Set}$ so we get $s[\llbracket t \rrbracket] : \llbracket \Gamma \rrbracket \rightarrow \llbracket \sigma \rrbracket$

for every term $\Gamma \vdash t : \sigma$. How does R_0 come in?

BACK TO STLC IN SET :

given $s: \text{Base} \rightarrow \text{Set}$ so we get $s[\llbracket \cdot \rrbracket]: \llbracket \Gamma \rrbracket \rightarrow \llbracket \sigma \rrbracket$
for every term $\Gamma \vdash t:\sigma$. How does R_σ come in?

DEFN: Pred is the category with

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

Pred is a ccc

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

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

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

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

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

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

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

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

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

Pred is a ccc

$$1 = (1, 1)$$

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

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

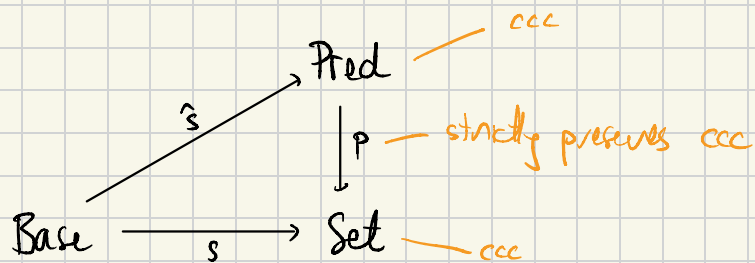
$$\text{where } f \in \bar{X} \supset \bar{Y} \Leftrightarrow \forall x \in \bar{X}. f x \in \bar{Y}$$

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

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

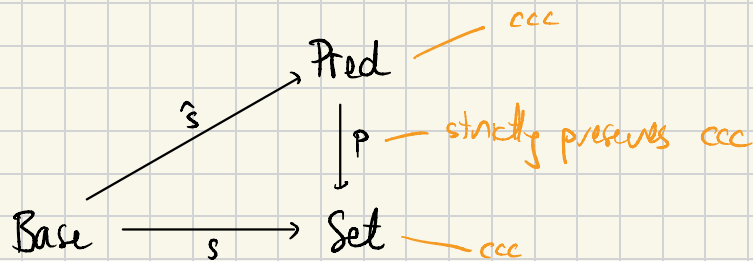
Suppose we pick for each $\beta \in \text{Base}$ a relation $R_\beta \subseteq \{\uparrow, \downarrow\}$.

This amounts to:

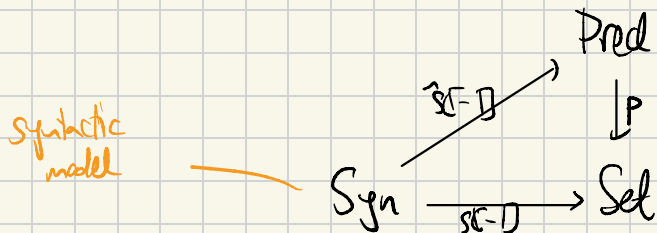


Suppose we pick for each $\beta \in \text{Base}$ a relation $R_\beta \subseteq \llbracket \beta \rrbracket$.

This amounts to:

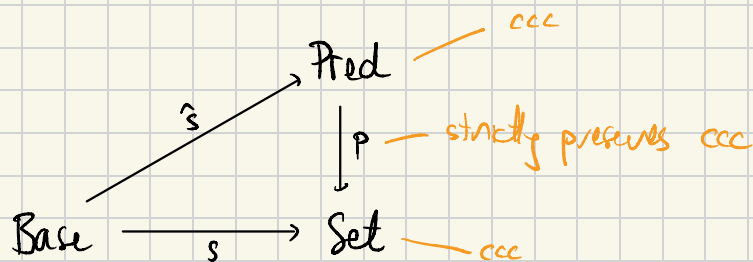


Hence, by induction / initiality, we get

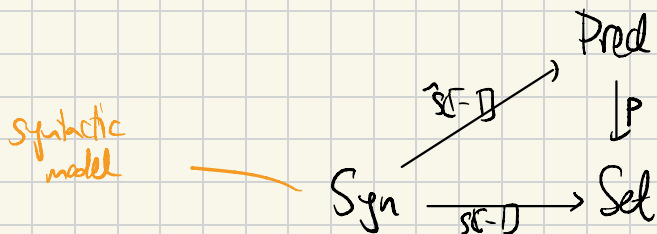


Suppose we pick for each $\beta \in \text{Base}$ a relation $R_\beta \subseteq \llbracket \beta \rrbracket$.

This amounts to:

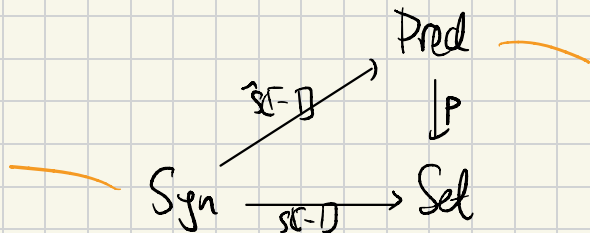


Hence, by induction / initiality, we get



$\hat{s}[\llbracket \cdot \rrbracket]$ encodes
a logical relation:
 $\hat{s}[\llbracket \cdot \rrbracket] = (s[\llbracket \cdot \rrbracket], R_0)$

synthetic
model



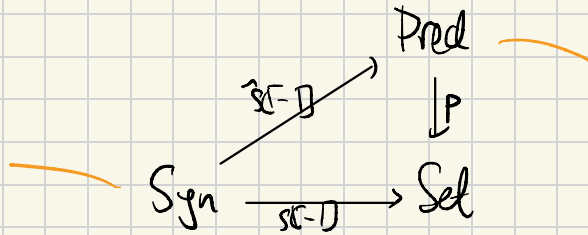
$$1 := (1, 1)$$

$$(x_1, \bar{x}_1) \times (x_2, \bar{x}_2) := (x_1 \times x_2, \bar{x}_1 \times \bar{x}_2)$$

$$(x, \bar{x}) \Rightarrow (y, \bar{y}) := (x \Rightarrow y, \bar{x} \supset \bar{y})$$

$$\text{where } f \in \bar{x} \supset \bar{y} \Leftrightarrow \forall x \in \bar{x}. f x \in \bar{y}$$

synthetic
model



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

$$(x_1, \bar{x}_1) \times (x_2, \bar{x}_2) := (x_1 \times x_2, \bar{x}_1 \times \bar{x}_2)$$

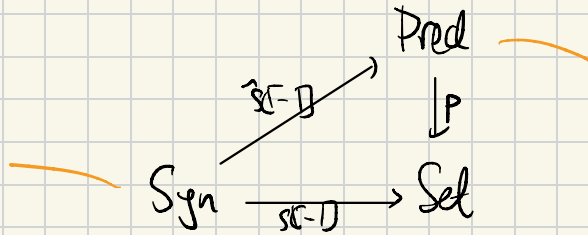
$$(x, \bar{x}) \Rightarrow (y, \bar{y}) := (x \Rightarrow y, \bar{x} \supset \bar{y})$$

$$\text{where } f \in \bar{x} \supset \bar{y} \Leftrightarrow \forall x \in \bar{x}. f x \in \bar{y}$$

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

$$\hat{s}[1] = (1, \{*\}) = (1, \underline{R_1})$$

synthetic
model



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

$$(x_1, \bar{x}_1) \times (x_2, \bar{x}_2) := (x_1 \times x_2, \bar{x}_1 \times \bar{x}_2)$$

$$(x, \bar{x}) \Rightarrow (y, \bar{y}) := (x \Rightarrow y, \bar{x} \supset \bar{y})$$

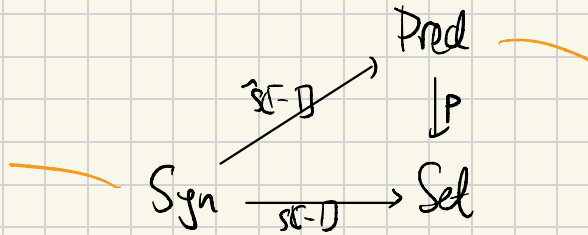
$$\text{where } f \in \bar{x} \supset \bar{y} \Leftrightarrow \forall x \in \bar{x}. f x \in \bar{y}$$

$$\hat{s}[f] = (s[f], \underline{R_f})$$

$$\hat{s}[1] = (1, \{*\}) = (1, \underline{R_1})$$

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

synthetic
model



$$1 := (1, 1)$$
$$(x_1, \bar{x}_1) \times (x_2, \bar{x}_2) := (x_1 \times x_2, \bar{x}_1 \times \bar{x}_2)$$

$$(x, \bar{x}) \Rightarrow (y, \bar{y}) := (x \Rightarrow y, \bar{x} \supset \bar{y})$$

$$\text{where } f \in \bar{x} \supset \bar{y} \Leftrightarrow \forall x \in \bar{x}. f x \in \bar{y}$$

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

$$\hat{s}[1] = (1, \{*\}) = (1, \underline{R_1})$$

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

$$\hat{s}[\sigma \rightarrow \tau] = \hat{s}[\sigma] \Rightarrow \hat{s}[\tau] = (s[\sigma], R_\sigma) \Rightarrow (s[\tau], R_\tau) = (s[\sigma] \Rightarrow s[\tau], \underline{R_\sigma \supset R_\tau})$$

SLOGAN: to give an STLC logical relation for a
[Reynolds, Ma] Model $(\mathbb{C}, \varepsilon)$ is to give

① a ccc \mathbb{E} of "relations"

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

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

$$\begin{array}{ccc} & \hat{s} & \mathbb{E} \\ & \nearrow & \downarrow p \\ \text{Base} & \xrightarrow{\varepsilon} & \mathbb{C} \end{array}$$

SLOGAN: to give an STLC logical relation for a
[Reynolds, Ma] Model $(\mathbb{C}, \varepsilon)$ is to give

① a a ccc \mathbb{E} of "relations"

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

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

$$\begin{array}{ccc} & \hat{s} & \mathbb{E} \\ & \nearrow & \downarrow p \\ \text{Base} & \xrightarrow{\varepsilon} & \mathbb{C} \end{array}$$

... then R_0 arises via $\hat{s}[\cdot]$ in \mathbb{E} .

SLOGAN: to give an STLC logical relation for a
[Reynolds, Ma] Model $(\mathbb{C}, \varepsilon)$ is to give

what
is this?

① a ccc \mathbb{E} of "relations"

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

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

$$\begin{array}{ccc} & \hat{s} & \mathbb{E} \\ & \nearrow & \downarrow p \\ \text{Base} & \xrightarrow{\varepsilon} & \mathbb{C} \end{array}$$

... then R_0 arises via $\hat{s}[\cdot]$ 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?

L as "fibrations for logical relations"

$P: \text{Pred} \rightarrow \text{Set}$ has a special property:

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

$P: \text{Pred} \rightarrow \text{Set}$ has a special property:

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

$$\begin{array}{ccc} & (Y, \bar{y}) & \text{Pred} \\ & & \downarrow P \\ X \xrightarrow{f} Y = p(Y, \bar{y}) & & \text{Set} \end{array}$$

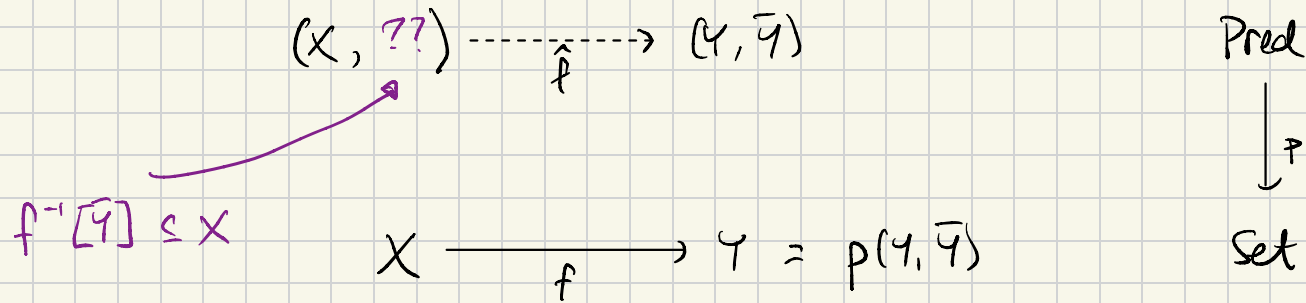
$P: \text{Pred} \rightarrow \text{Set}$ has a special property:

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

$$\begin{array}{ccc} (X, ??) & \xrightarrow{\hat{f}} & (Y, \bar{y}) \\ & & \text{Pred} \\ & & \downarrow P \\ X & \xrightarrow{f} & Y = p(Y, \bar{y}) \\ & & \text{Set} \end{array}$$

$P: \text{Pred} \rightarrow \text{Set}$ has a special property:

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



(Grothendieck)

DEFⁿ: a [^]fibration is a functor $p : E \rightarrow B$ such that for every $f : A \rightarrow pY$ in B there is an object $\hat{A} \in E$ and a map $\hat{f} : \hat{A} \rightarrow Y$ st.

- $p(\hat{A}) = A$

- $p(\hat{f}) = f$

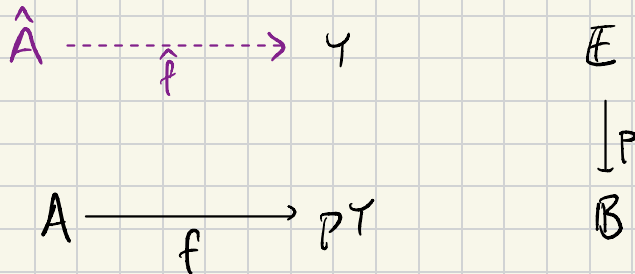
- a universal property holds

(Grothendieck)

DEFⁿ: a fibration is a functor $p: E \rightarrow B$ such that for every $f: A \rightarrow pY$ in B there is an object $\hat{A} \in E$ and a map $\hat{f}: \hat{A} \rightarrow Y$ st.

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

PICTORIALLY:



EXAMPLES

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

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

• $\text{cod}: \mathbb{C}^{\rightarrow} \longrightarrow \mathbb{C}$ the codomain fibration

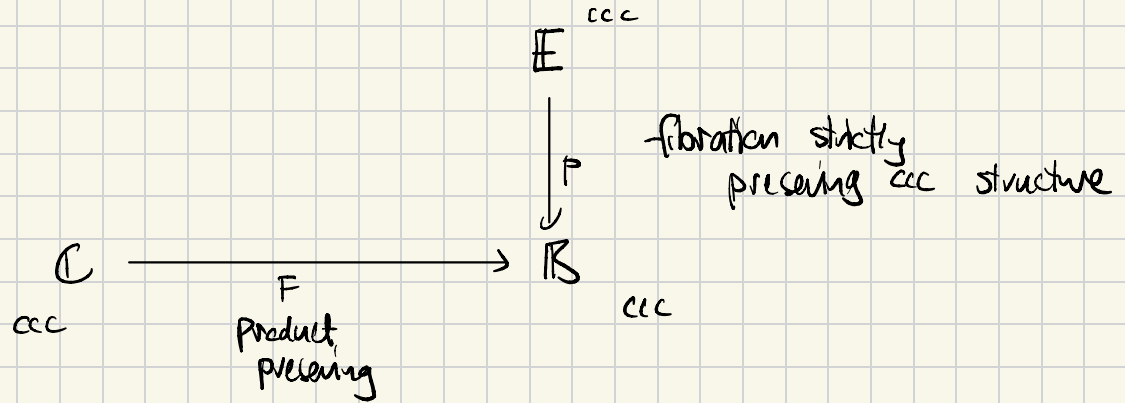
obj: $f: X \longrightarrow Y$ in \mathbb{C}

maps: squares
$$\begin{array}{ccc} X & \longrightarrow & X' \\ p \downarrow & \cong & \downarrow p' \\ Y & \longrightarrow & Y' \end{array}$$

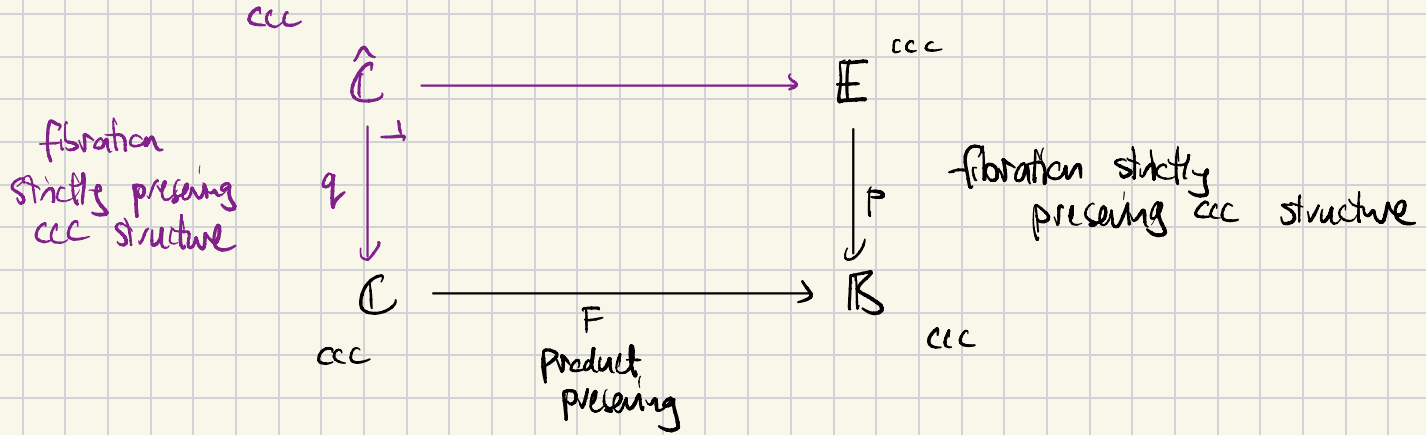
A RECIPE :

C
cc

A RECIPE :

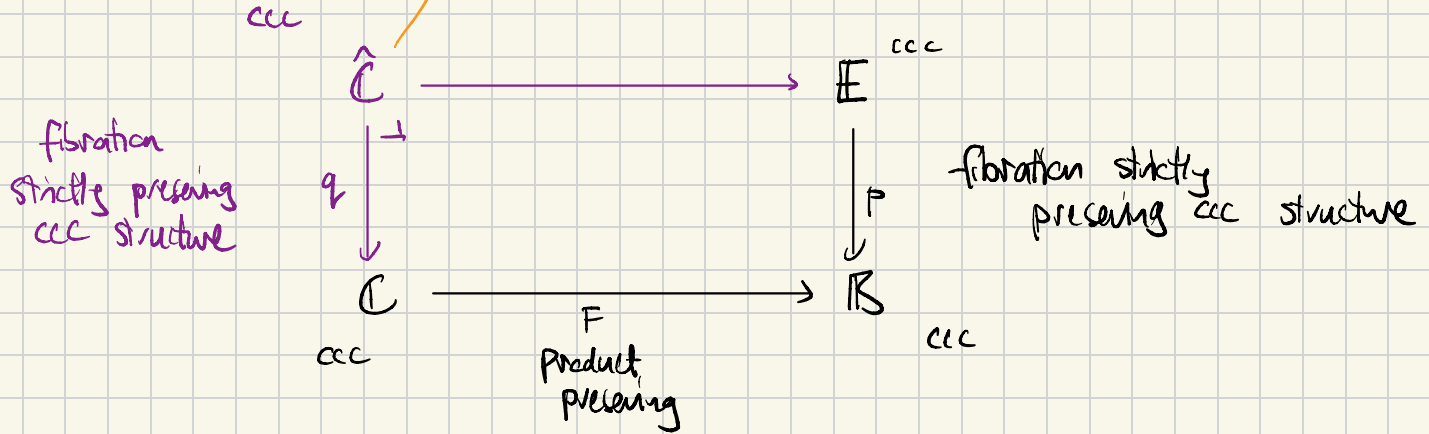


A RECIPE :



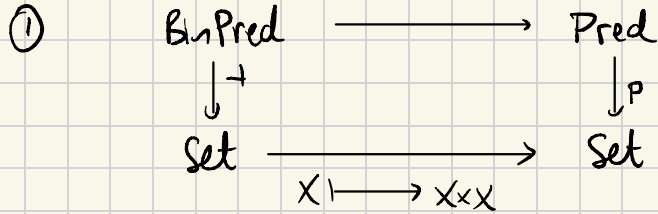
A RECIPE :

obj: $(C \in \mathcal{C}, X \in E)$ st. $FX = PX$,
i.e. a \mathcal{C} -object and a 'relation' on it

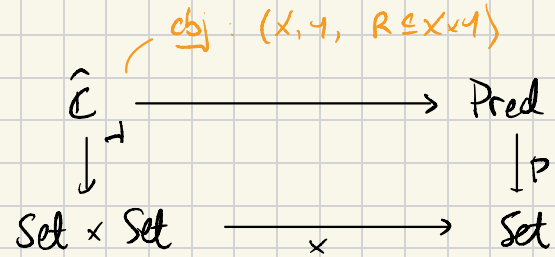


EXAMPLES

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

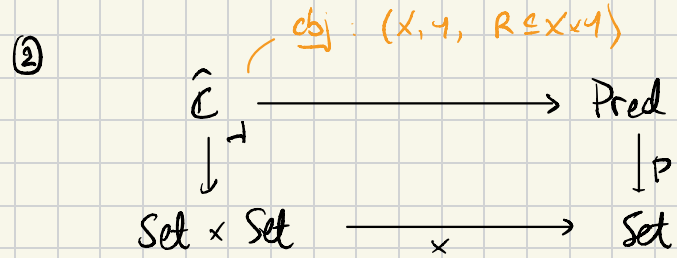
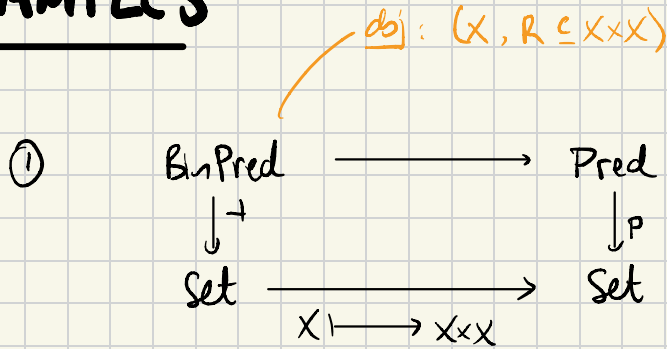


②

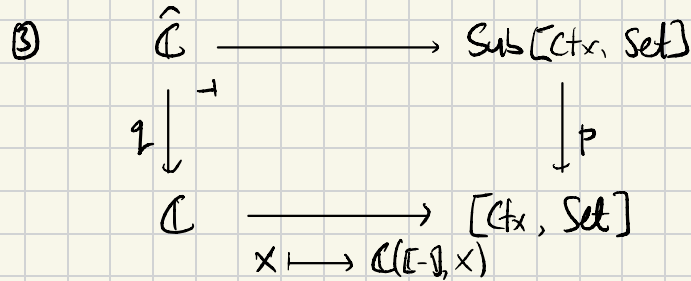


can start to
relate models

EXAMPLES



can start to
relate models



obj: $(X \in C, R \mapsto C(\Gamma, X))$

ie. for every context Γ a subset

$R(\Gamma) \subseteq C(\Gamma, X)$,

compatible with remainings

Kripke relations of varying arity

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 — i.e. 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 — i.e. 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 — i.e. the model
- ③ the general theory gives us a useful framework for finding such fibrations (= relations models)

DEFN : an 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?

L 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 $\tau ::= \dots \mid T\tau$

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

a pure program
is trivially effectful

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

run u , bind it to x , then run t .

Haskell notation:

$t \gg x$

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 : $\mathcal{CC} \subseteq \mathcal{C} + \text{a (strong) monad } T$

$$\llbracket T\sigma \rrbracket := T\llbracket\sigma\rrbracket$$

$$\llbracket \text{return } t \rrbracket := \llbracket \Gamma \rrbracket \xrightarrow{\mathcal{C}t} \llbracket \sigma \rrbracket \xrightarrow{\eta} T\llbracket\sigma\rrbracket$$

SEMANTICS : $\text{ccc } \mathbb{C} + \text{a (strong) Monad } T$

$$\llbracket T\sigma \rrbracket = T\llbracket\sigma\rrbracket$$

$$\llbracket \text{return } t \rrbracket := \llbracket \Gamma \rrbracket \xrightarrow{\llbracket t \rrbracket} \llbracket \sigma \rrbracket \xrightarrow{1} T\llbracket\sigma\rrbracket$$

eg

for powerset : $\llbracket \text{return } t \rrbracket(\gamma) = \{\llbracket t \rrbracket(\gamma)\}$

for Maybe : $\llbracket \text{return } t \rrbracket(\gamma) = \text{inl}(\llbracket t \rrbracket(\gamma)) \in \llbracket \sigma \rrbracket + 1$

for Writer : $\llbracket \text{return } t \rrbracket(\gamma) = (\llbracket t \rrbracket(\gamma), \varepsilon)$

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

$$\llbracket T\sigma \rrbracket := T\llbracket \sigma \rrbracket$$

$$\llbracket \text{return } t \rrbracket := \llbracket \Gamma \rrbracket \xrightarrow{\llbracket t \rrbracket} \llbracket \sigma \rrbracket \xrightarrow{1} T\llbracket \sigma \rrbracket$$

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

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

$$\llbracket T\sigma \rrbracket = T\llbracket \sigma \rrbracket$$

$$\llbracket \text{return } t \rrbracket := \llbracket \Gamma \rrbracket \xrightarrow{\llbracket t \rrbracket} \llbracket \sigma \rrbracket \rightarrow T\llbracket \sigma \rrbracket$$

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

eg. for powerset : $\llbracket u \rrbracket(\gamma) \in \mathcal{P}\llbracket \sigma \rrbracket$ and $\llbracket t \rrbracket : \llbracket \Gamma \rrbracket \times \llbracket \sigma \rrbracket \rightarrow \mathcal{P}\llbracket \tau \rrbracket$, so

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

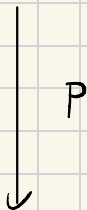
for writers:

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

strict

MAPS OF λ_{MI} MODELS

cc +
strong monad (E, \hat{T})

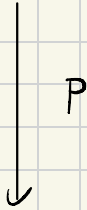


cc +
strong monad (B, T)

strict

MAPS OF λ_{MI} MODELS

ccc +
strong monad (E, \hat{T})



ccc +
strong monad (B, T)

P preserves ccc structure
+ monad structure on the nose:

$$P(\hat{T}X) = T(PX)$$

$$P(\hat{\mu}_X) = \mu_{PX}$$

$$P(\hat{\eta}_X) = \eta_{PX}$$

$$P(\hat{\sigma}_{X,Y}) = \sigma_{P_X, P_Y}$$

EXAMPLE:

1) If E, B are ccs and $p: E \rightarrow B$

preserves ccc-structure, $(E, (- \Rightarrow R) \Rightarrow R) \xrightarrow{p} (B, (- \Rightarrow pR) \Rightarrow pR)$

EXAMPLE:

1) if E, B are ccs and $p: E \rightarrow B$

preserves ccc-structure, $(E, (- \Rightarrow R) \Rightarrow R) \xrightarrow{p} (B, (- \Rightarrow pR) \Rightarrow pR)$

2) if $C \in \text{Set}$ is a set of characters and $\bar{C} \subseteq C$

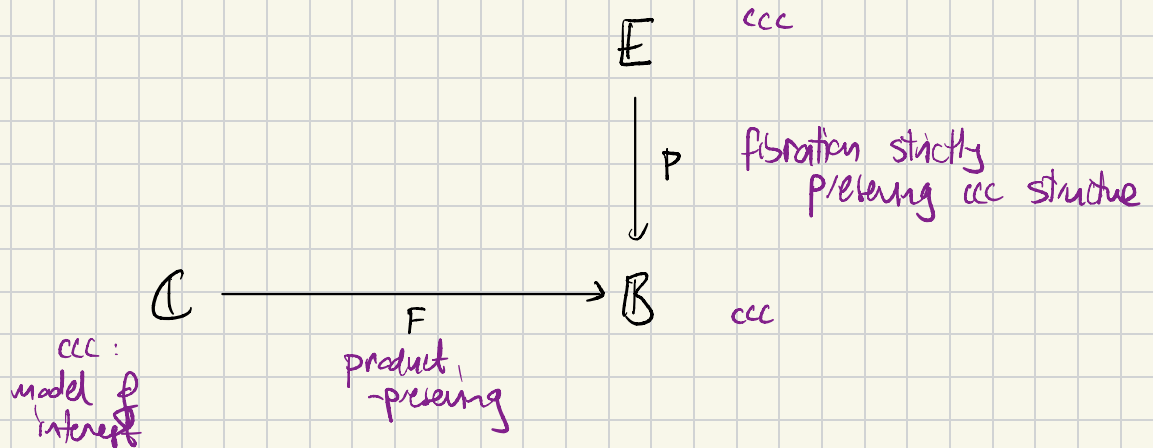
then $\bar{C}^* \xrightarrow{\text{submonoid}} C^*$ so $(-) \times (C^*, \bar{C}^*) : \text{Pred} \rightarrow \text{Pred}$
 $(x, \bar{x}) \longmapsto (x \times C^*, \bar{x} \times \bar{C}^*)$

and

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

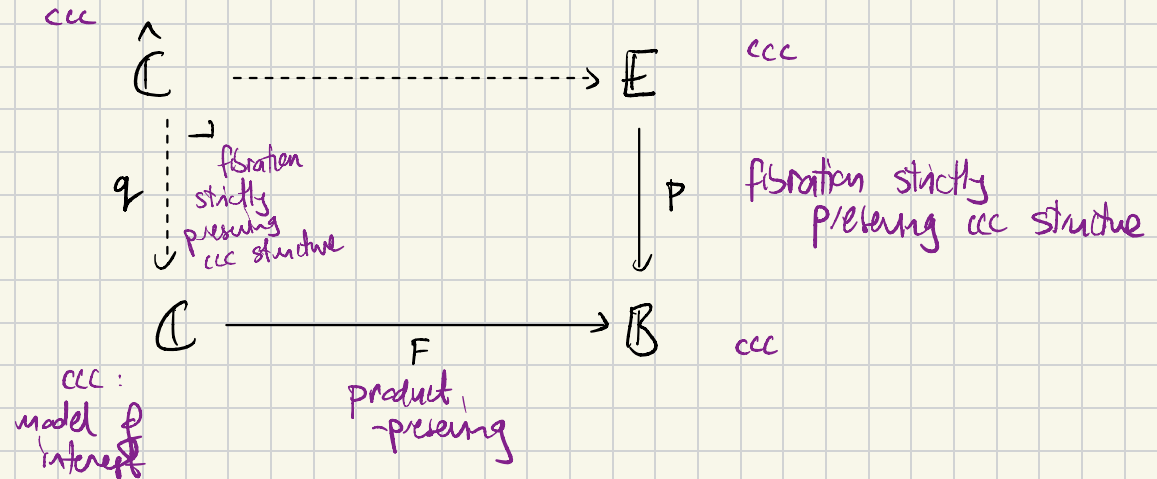
LOGICAL RELATIONS FOR λ_{ml}

Take the STLC picture...



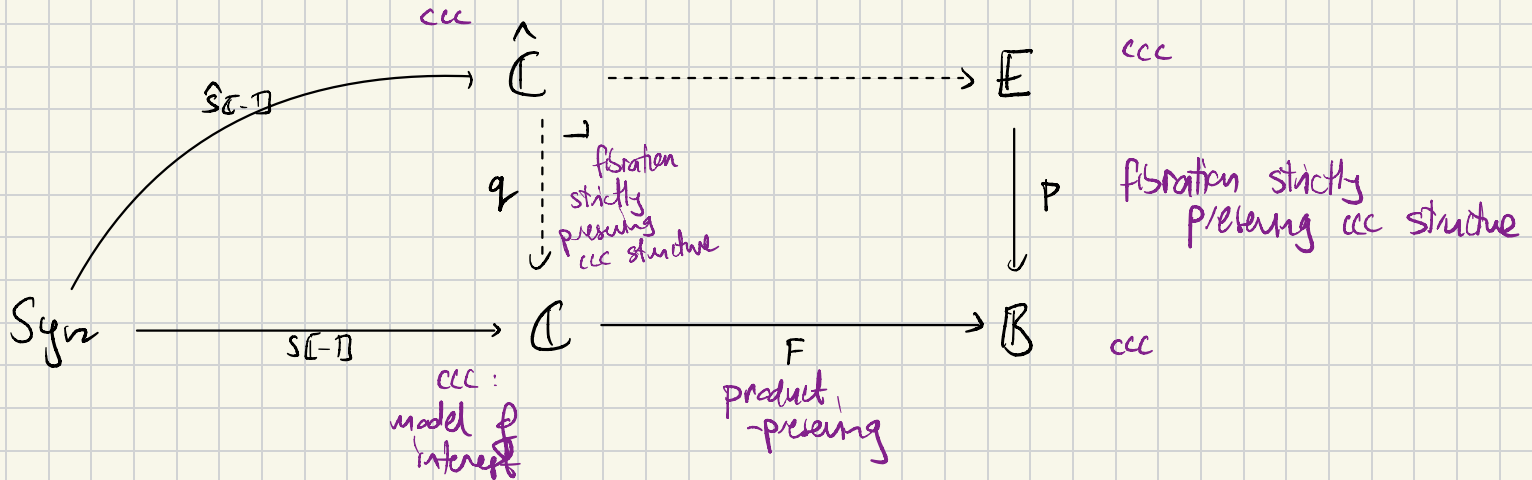
LOGICAL RELATIONS FOR λ_{ml}

Take the STLC picture...



LOGICAL RELATIONS FOR λ_{ml}

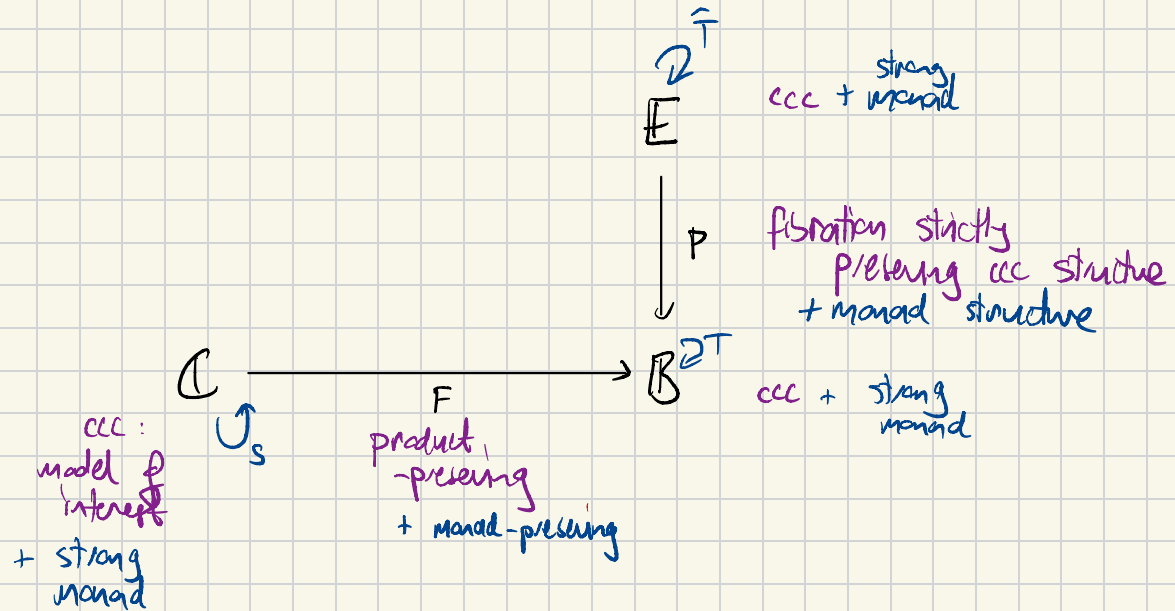
Take the STLC picture...



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

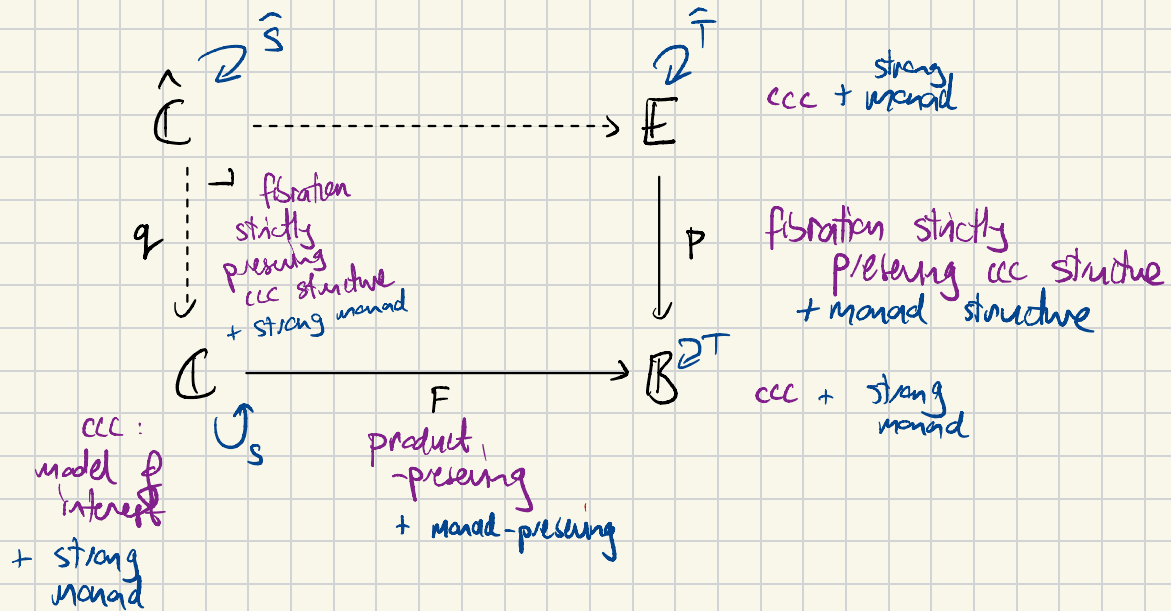
LOGICAL RELATIONS FOR λ_{ml}

Take the STLC picture... and add monads



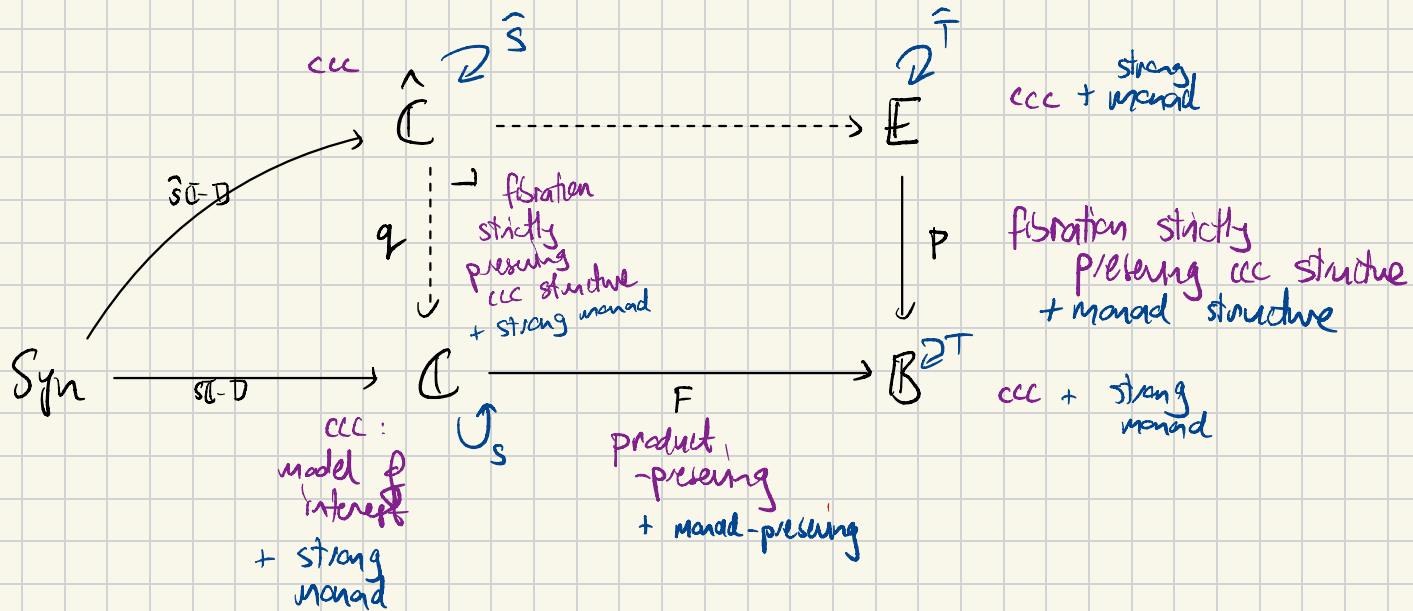
LOGICAL RELATIONS FOR λ_{ml}

Take the STLC picture... and add monads



LOGICAL RELATIONS FOR λ_{ml}

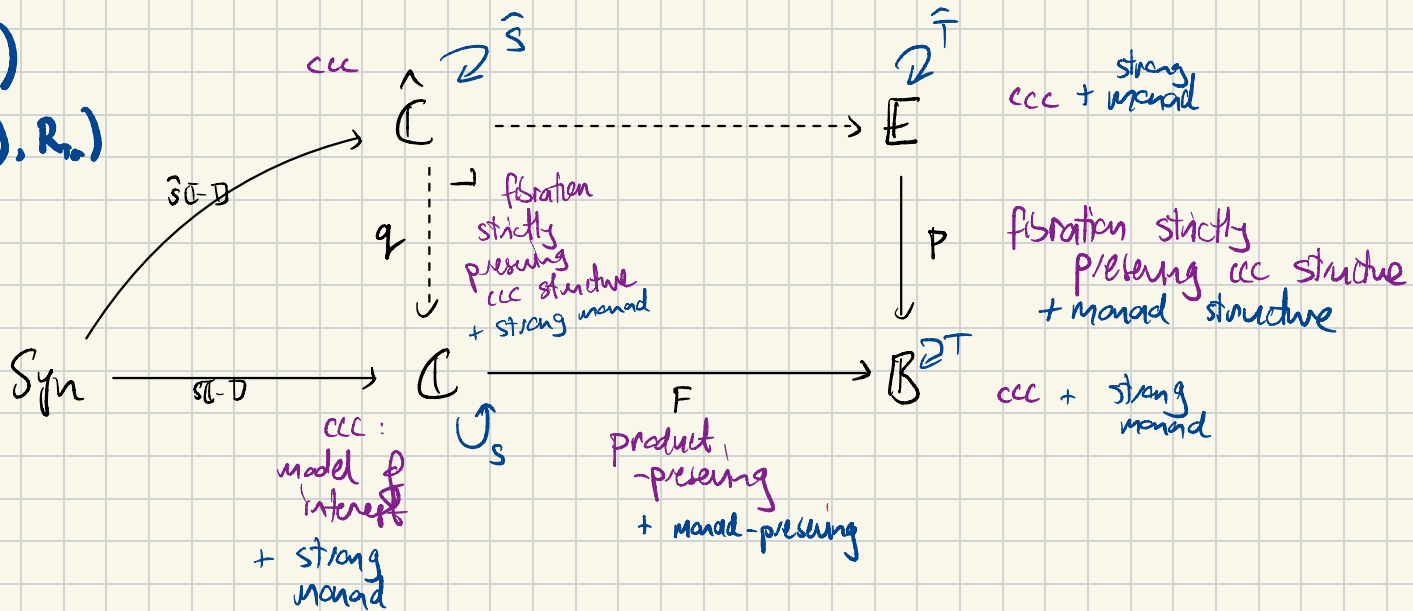
Take the STLC picture... and add monads



LOGICAL RELATIONS FOR λ_{ml}

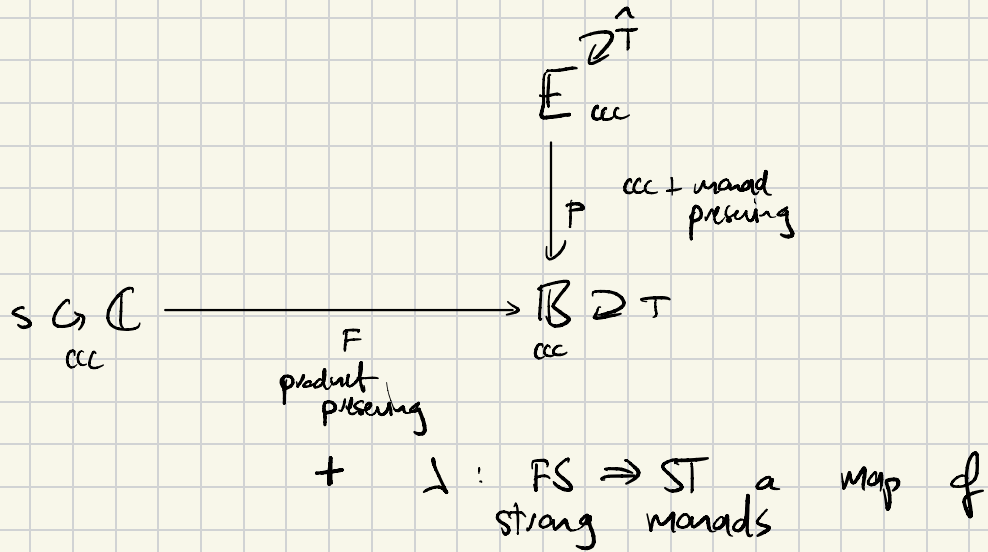
Take the STLC picture... and add monads

$$\hat{S}(s\mathbb{E}\text{-D}, R_G) = (S(s\mathbb{E}\text{-D}), R_G)$$



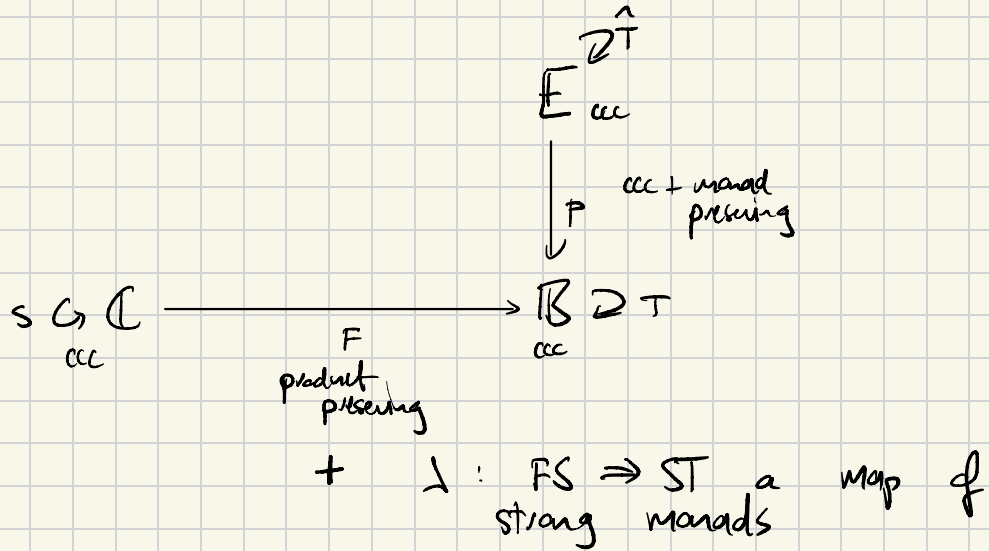
Prop^w. [Katsumata, Katsumata - Kammar - S.]

Suppose you have



Propn. [Katsumata, Katsumata - Kammar - S.]

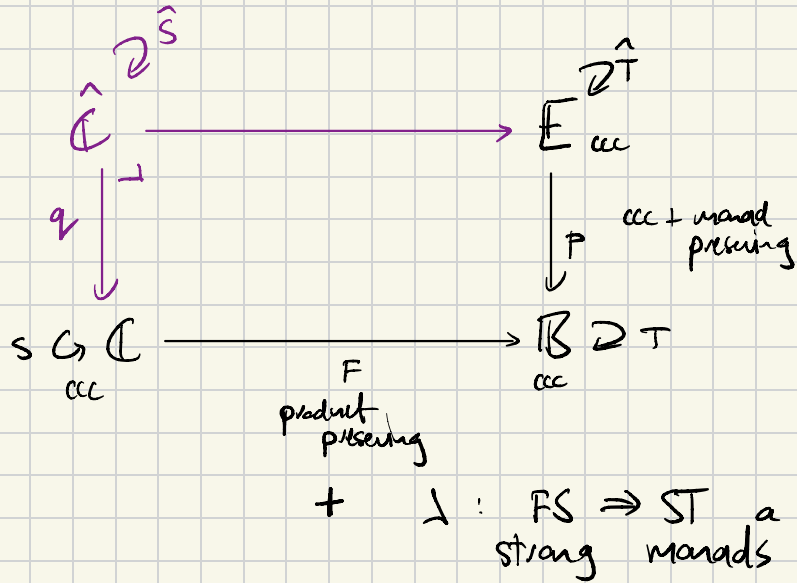
Suppose you have



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

Prop^w. [Katsumata, Katsumata - Kammar - S.]

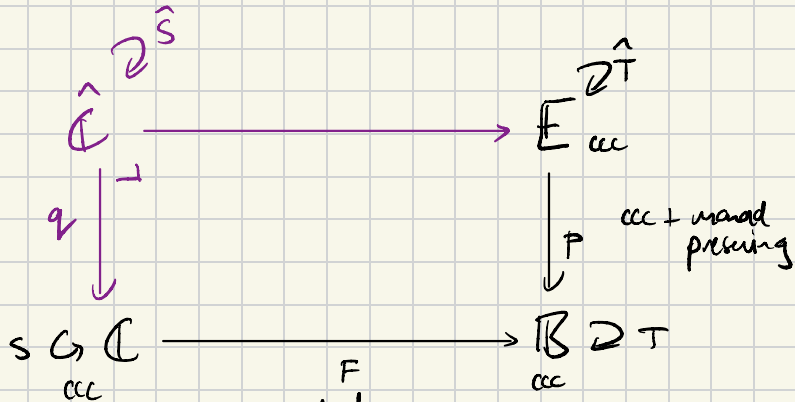
Suppose you have



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

Prop^w. [Katsumata, Katsumata - Kammar - S.]

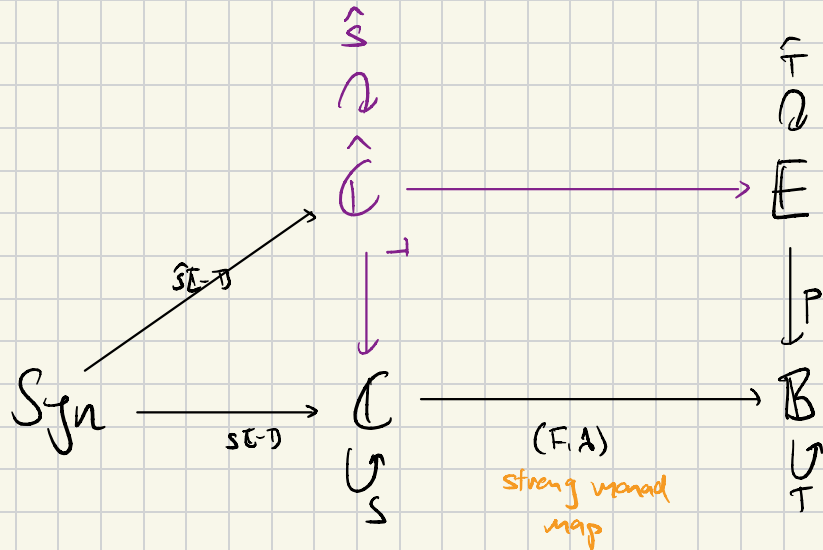
Suppose you have



this is a fibration!

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

DEFINING A LOGICAL RELATION



ie. we have

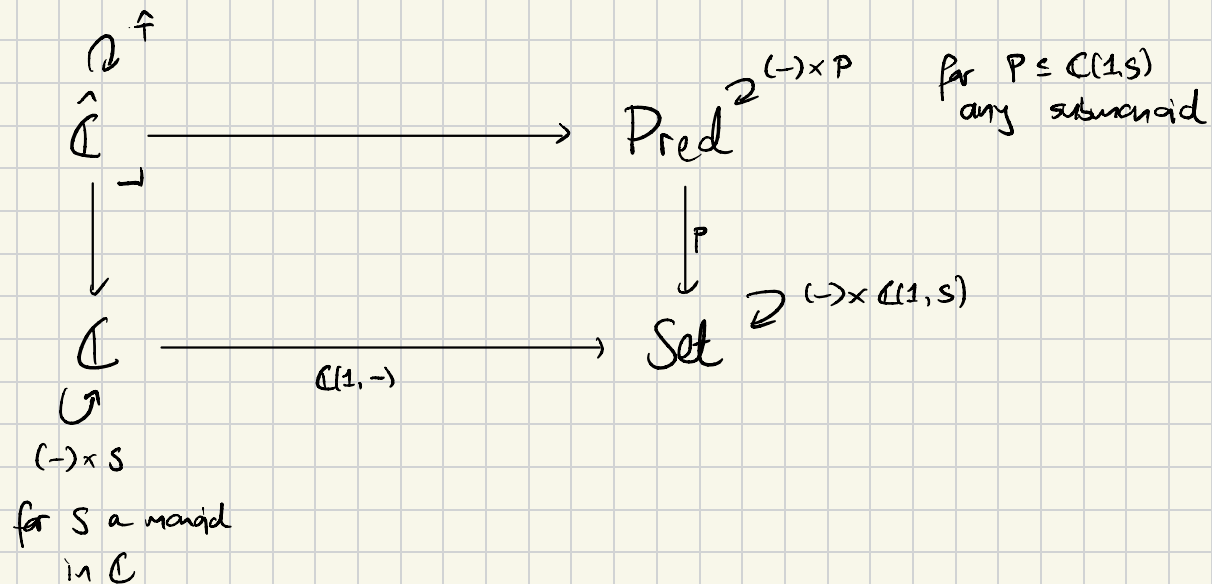
$$\hat{S}[\sigma] = (s[\sigma], R_\sigma)$$

so we get

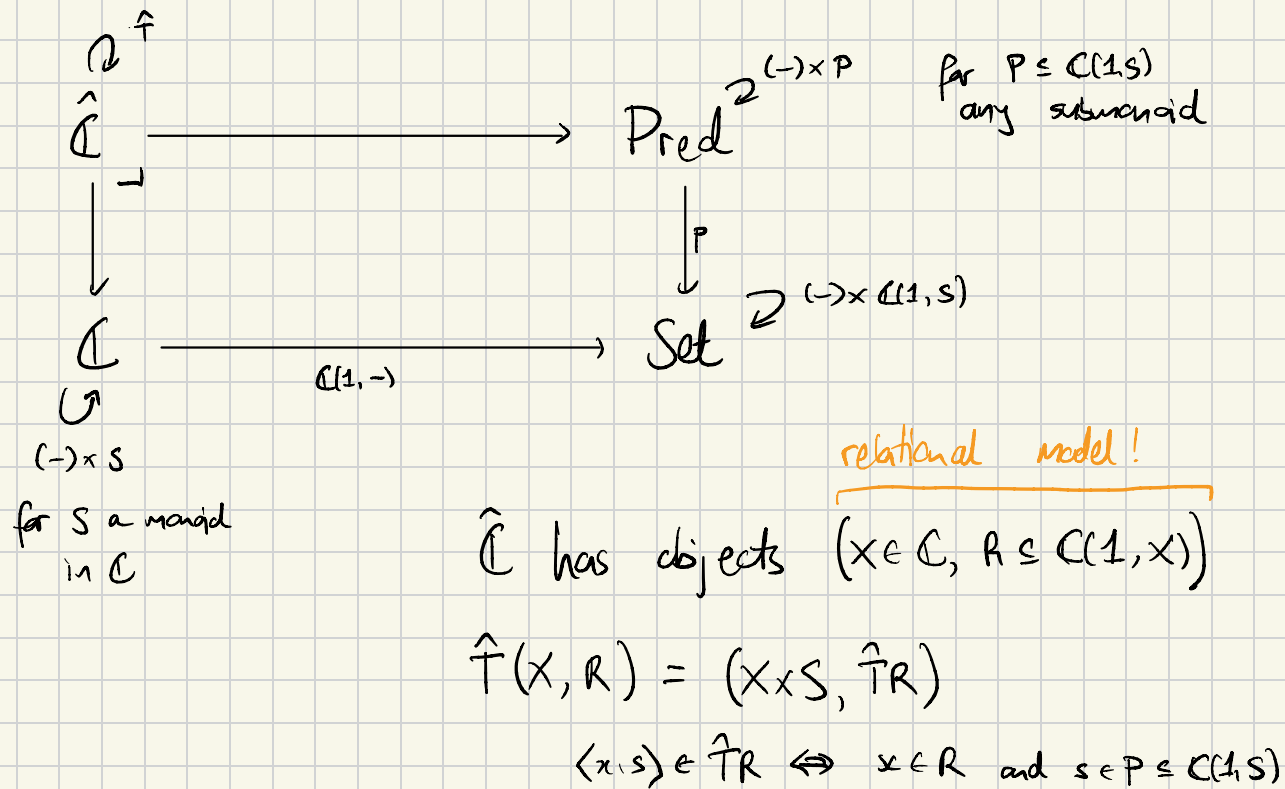
$$R = \{ R_\sigma \mid \sigma \in \text{Type} \}$$

including $T\tau$.

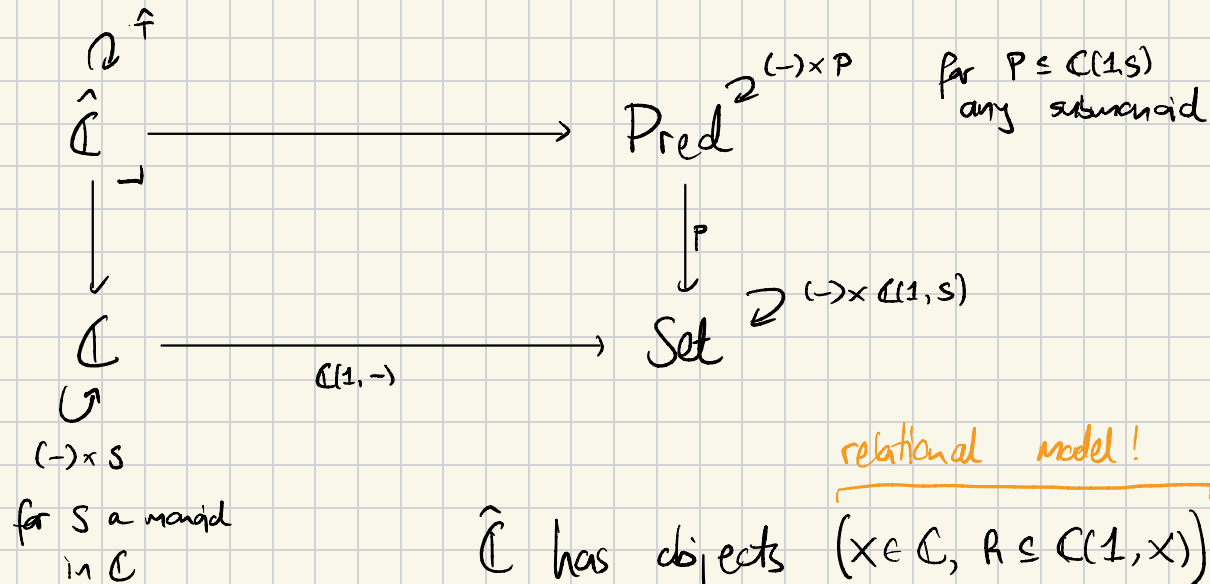
EXAMPLE : GLOBAL STATE



EXAMPLE : GLOBAL STATE



EXAMPLE : GLOBAL STATE



for a logical relation:

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

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

$$(x, s) \in \hat{T}R \Leftrightarrow x \in R \text{ and } s \in P \subseteq \mathcal{C}(1, S)$$

EXAMPLE : SIMULATING EFFECTS

à la Katsunata

STRONG MONAD MAP

$$\gamma : L \longrightarrow \mathcal{P}^{\text{fin}}$$

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

EXAMPLE : COMPARING EFFECTS

à la Katsunata

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

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

$$\begin{array}{c} \text{Set} \times \text{Set} \\ \cup \\ L \times \mathcal{P}_{fin} \end{array}$$

EXAMPLE : COMPARING EFFECTS

à la Katsunata

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

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

 $\hat{\tau}$ \Downarrow

BinPred

$$\underline{\text{def}}: (X, Y, R \Rightarrow X \times Y)$$

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

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

 $\downarrow P$

$$\begin{array}{c} \text{Set} \times \text{Set} \\ \cup \\ L \times \mathcal{P}_{fin} \end{array}$$

$$\xrightarrow{(\text{id}, \gamma \times \text{id})}$$

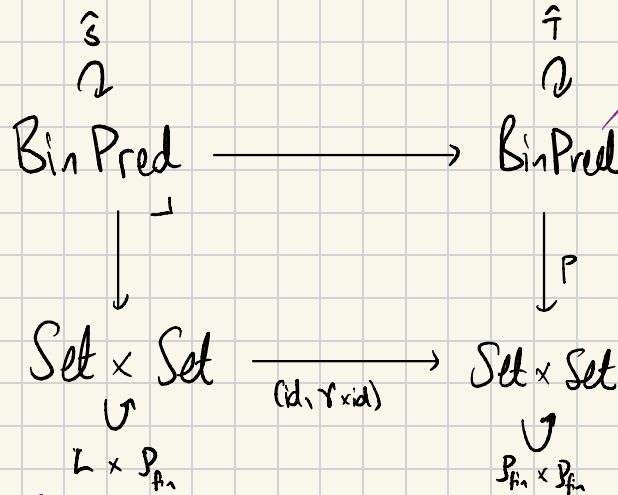
$$\begin{array}{c} \text{Set} \times \text{Set} \\ \cup \\ \mathcal{P}_{fin} \times \mathcal{P}_{fin} \end{array}$$

EXAMPLE : COMPARING EFFECTS

à la Katsumata

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

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



$$\hat{\gamma}: (X, Y, R) \rightsquigarrow X \times Y$$

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

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

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

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

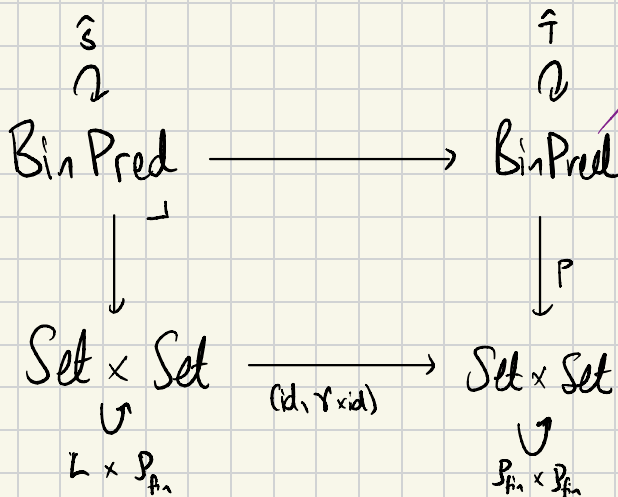
$$\Leftrightarrow n=m \text{ and } (x_i, y_i) \in R \text{ for } i=1, \dots, n$$

EXAMPLE : COMPARING EFFECTS

à la Katsumata

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

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



$$\hat{\sigma}: (X, Y, R) \mapsto X \times Y$$

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

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

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

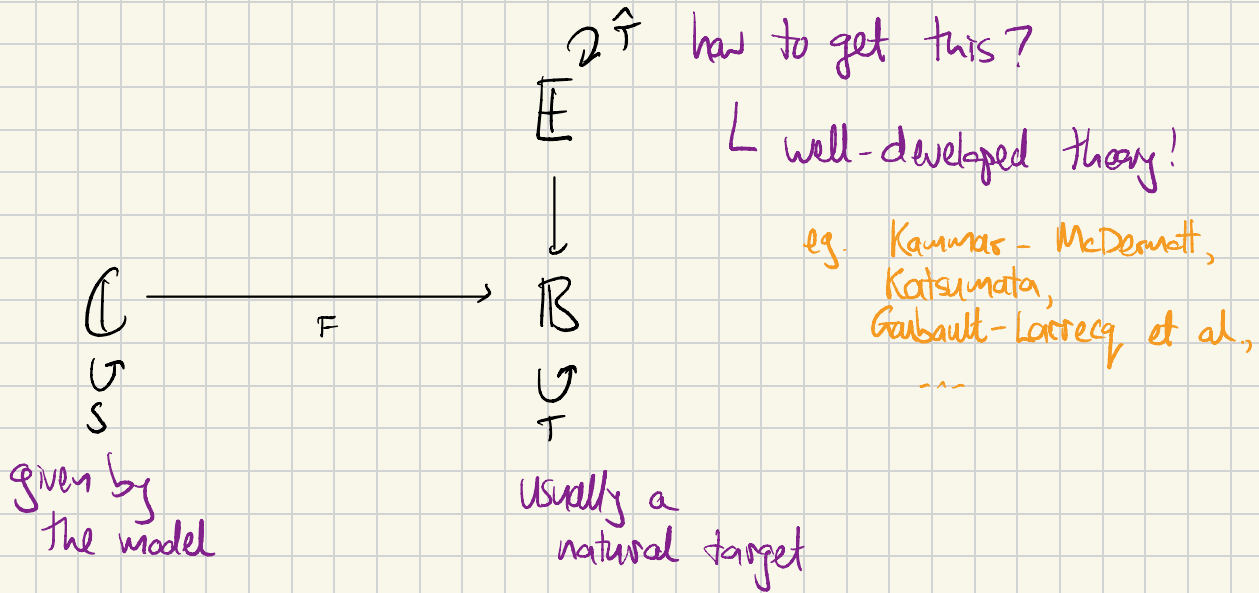
$$([x_1, \dots, x_n], \{y_1, \dots, y_m\}) \in \hat{S}R$$

$$\Leftrightarrow n=m \text{ and } (x_i, y_i) \in R \text{ for } i=1, \dots, n$$

LOGICAL RELATION M :

$$([x_1, \dots, x_n], \{y_1, \dots, y_m\}) \in M_{\gamma \circ \text{id}} \Leftrightarrow m=n \text{ and each } (x_i, y_i) \in M_0$$

DIFFICULT BIT: defining the target lifting



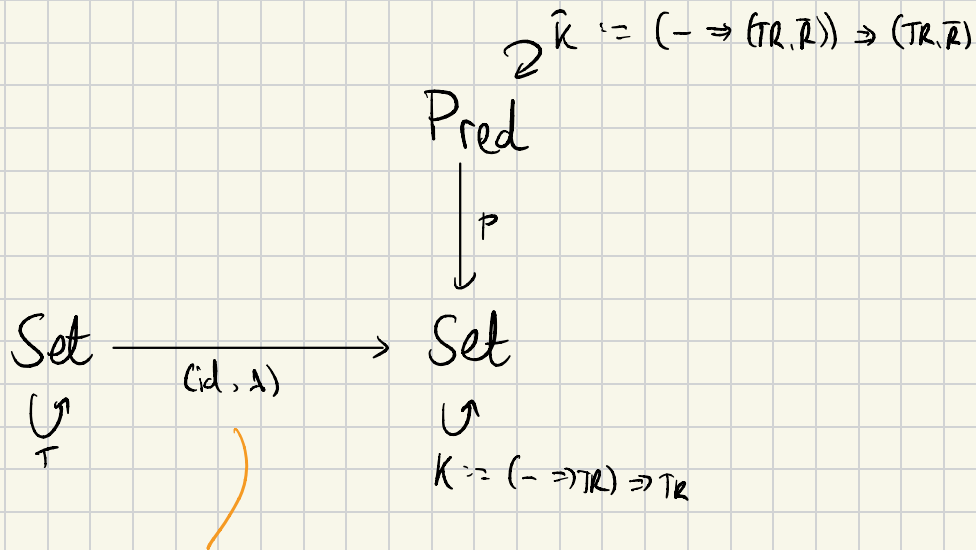
EXAMPLE : TT-LIFTING (Katsumata)

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

Set
 \cup
T

EXAMPLE : TT-LIFTING (Kotsumata)

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

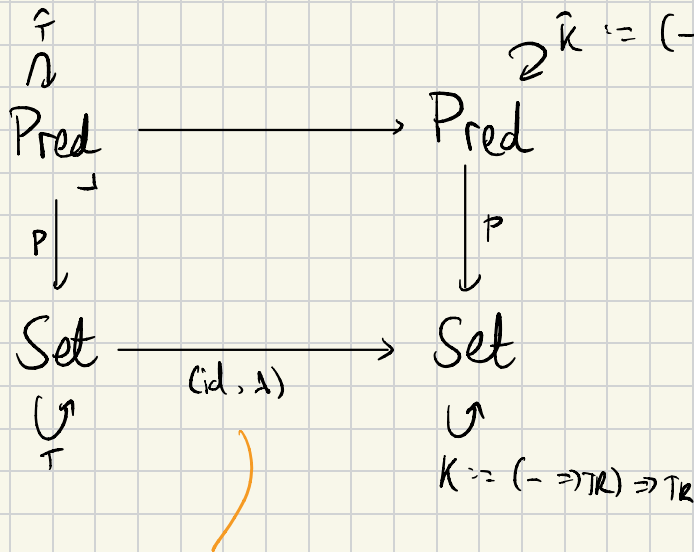


$$\lambda_x : TX \longrightarrow (X \Rightarrow TR) \Rightarrow TR$$

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

EXAMPLE : TT-LIFTING (Kotsumata)

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



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

$$t \in \hat{\text{TX}} \Leftrightarrow \forall f \in (\bar{X} \Rightarrow \bar{R}).$$

let $x = t$ in $f x \in \bar{R}$

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

cf. Krivine realisability

$$\lambda_x : \text{TX} \longrightarrow (X \Rightarrow \text{TR}) \Rightarrow \text{TR}$$

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

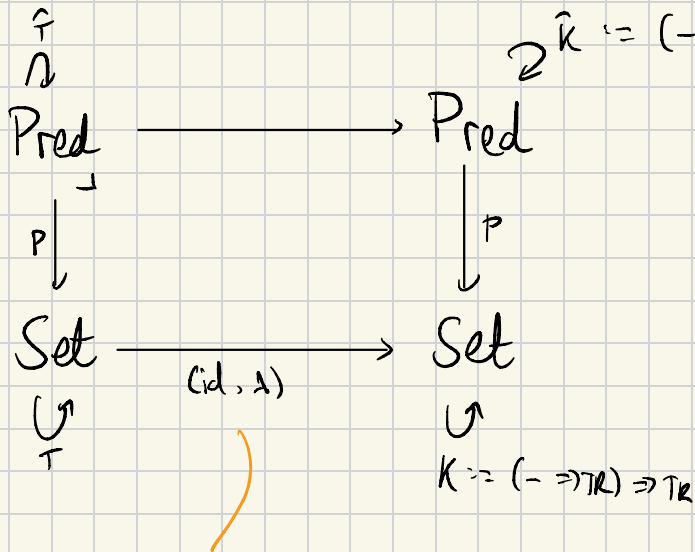
EXAMPLE : TT-LIFTING (Kotsumata)

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

LOGICAL RELATION S :

$$t \in S_{\text{TR}} \Leftrightarrow \forall f \in (S_{\sigma} \Rightarrow \bar{R}).$$

let $x = t$ in $f x \in \bar{R}$



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

$$t \in \hat{\text{TX}} \Leftrightarrow \forall f \in (\bar{X} \Rightarrow \bar{R}).$$

let $x = t$ in $f x \in \bar{R}$

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

cf. Krivine realisability

$$\lambda_x : \text{TX} \longrightarrow (X \Rightarrow \text{TR}) \Rightarrow \text{TR}$$

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

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 — i.e. 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 CBV fibration is a fibration that strictly preserves CCC + the monad \approx a logical relation

④ How does this picture
extend to CBPV?

Why CBPV?

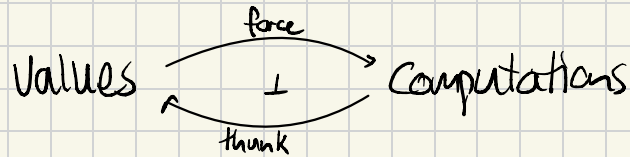
Subsumes both CBV and CBN

-- by giving fine control over

when effects happen

FEATURES OF CFPV

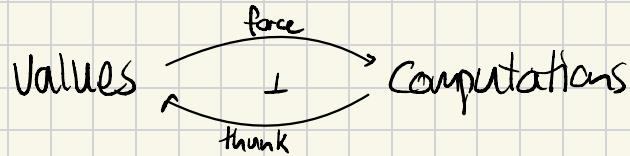
- Computations and values separate, but related:



} no "accidental" reductions - contract to λ

FEATURES OF CBN

- Computations and values separate, but related:

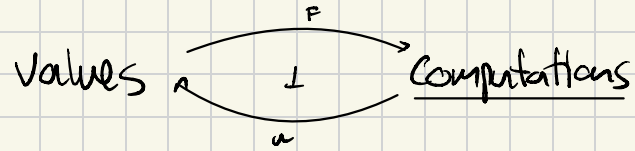


} no "accidental" reductions - contract to λ

best of both worlds

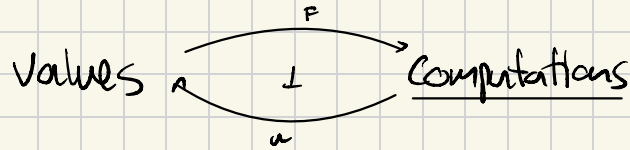
- "sums are nice in CBN" + "arrows are nice in CBN"
 - ie. the sum type consists of things like $\text{inj}_i V$ for V a value
 - in CBN, effects commute with λ so the η -law holds

Some syntax



"computations do,
values are"

Some syntax



"computations do,
values are"

contexts are values

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

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

prevent a computation
running

$$\Gamma \vdash^v V : \underline{B}$$

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

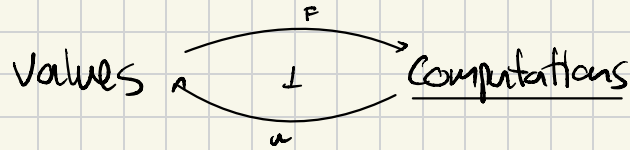
allow the
computation to
run

$$\Gamma \vdash^c V : A$$

$$\Gamma \vdash^c \text{return } V : \underline{A}$$

values are trivial
computations

Some syntax



"computations do,
values are"

contexts are values

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

values are trivial
computations

$$\frac{\overbrace{\Gamma \vdash M : B}}{\Gamma \vdash \text{thunk } M : \underline{UB}}$$

prevent a computation
running

$$\frac{\Gamma \vdash V : \underline{UB}}{\Gamma \vdash \text{force } V : B}$$

allow the
computation to
run

CBV function type becomes

$$\overline{A \rightarrow_{\text{cbv}} B} = U(A \rightarrow B)$$

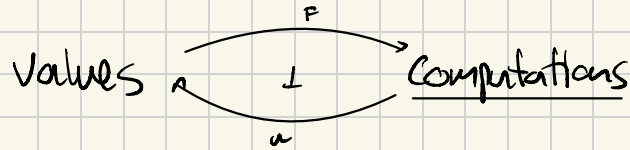
function types
are computations

$$\overline{\lambda x. M} = \text{thunk } \lambda x. \overline{M}$$

value : cbv

makes it a value

Some syntax



"computations do,
values are"

contexts are values

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

values are trivial
computations

$$\frac{\overbrace{\Gamma \vdash M : B}}{\Gamma \vdash \text{thunk } M : \underline{UB}}$$

prevent a computation
running

$$\frac{\Gamma \vdash V : \underline{UB}}{\Gamma \vdash \text{force } V : B}$$

allow the
computation to
run

CBV function type becomes

$$\overline{A \rightarrow_{\text{CBV}} B} = U(A \rightarrow \underline{B})$$

function types
are computations

$$\overline{\lambda x. M} = \underline{\text{thunk } \lambda x. \overline{M}}$$

value : CBV

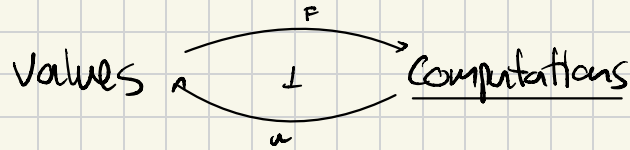
makes it a value

CBN function type becomes

$$\overline{A \rightarrow_{\text{CBN}} B} = (\underline{UA}) \rightarrow B$$

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

Some syntax



"computations do,
values are"

contexts are values

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

$$\Gamma \vdash^v \text{thunk } M : \underline{UB}$$

prevent a computation
running

$$\Gamma \vdash^v V : \underline{UB}$$

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

allow the
Computation to
run

$$\Gamma \vdash^c V : A$$

$$\Gamma \vdash^c \text{return } V : \underline{FA}$$

values are trivial
computations

the monad
is still
there

$$\Gamma \vdash^c V : A$$

$$\Gamma \vdash^c \text{return } V : \underline{FA}$$

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

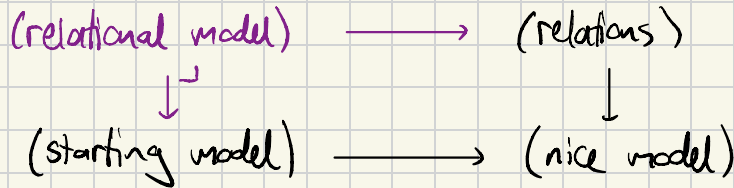
What we want:

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



3) recover sensible syntactic defⁿ

$\{R_A \mid A \in \text{ValType}\}$
 $\{R_B \mid B \in \text{CompType}\}$

Semantics of CRPV [Levy]

Need to interpret

$\Gamma \Vdash V : A$
values

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

$\Gamma \mid \underline{B} \vdash^k k : \underline{C}$
stacks / contexts
cf. $\llbracket M \rrbracket$ for a λ -term M
and context $\llbracket - \rrbracket$

Semantics of CRPV [Levy]

Need to interpret

$\Gamma \Vdash V : A$
values

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

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

Need hints depending
on $\Gamma \in \text{Val}$ and $\underline{B}, \underline{C} \in \text{Comp}$

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

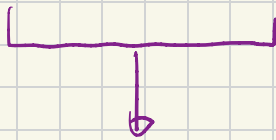
cf. $\llbracket M \rrbracket$ for a λ -term M
and context $\llbracket _ \rrbracket$

Semantics of CRPV [Levy]

Need to interpret

$$\Gamma \Vdash V : A$$

values



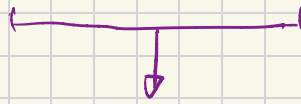
in a cartesian
category \mathcal{V} , ie.
in $\mathcal{V}(\llbracket \Gamma \rrbracket, \llbracket A \rrbracket)$

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



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

computations



$$\text{in } \mathcal{V}(\llbracket \Gamma \rrbracket, \llbracket \underline{B} \rrbracket) \\ \cong \mathcal{C}_{\text{arr}}(F1, \llbracket \underline{B} \rrbracket)$$

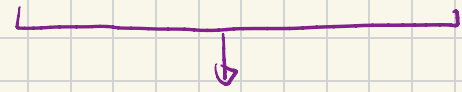
Need horns depending
on $\Gamma \in \text{Val}$ and $\underline{B}, \underline{C} \in \text{Comp}$



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

stacks / contexts

cf. $\llbracket \llbracket m \rrbracket \rrbracket$ for a λ -term m
and context $\llbracket - \rrbracket$



in a locally \mathcal{V} -indexed
category \mathcal{C} , ie. in
 $\mathcal{C}_{\text{arr}}(\llbracket \underline{B} \rrbracket, \llbracket \underline{C} \rrbracket)$

Semantics of CRPV [Levy]

DEFⁿ: let \mathbb{V} be cartesian. A locally \mathbb{V} -indexed category \mathcal{C} has

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

...

Semantics of CRPV [Levy]

DEF^m: let \mathbb{V} be cartesian. A locally \mathbb{V} -indexed
category \mathcal{C} has

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

st. each \mathcal{C}_Γ is a category with the same objects,
compatible with maps in \mathbb{V} //

Semantics of CRPV [Levy]

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

DEFⁿ: let \mathbb{V} be cartesian. A locally \mathbb{V} -indexed
category \mathcal{C} has

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

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

EXAMPLES for \mathbb{W} cartesian

- self \mathbb{W} has objects as in \mathbb{W} and

$$(\text{self } \mathbb{W})_r (A, B) := \mathbb{W}(\Gamma \times A, B)$$

EXAMPLES for \mathbb{V} cartesian

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

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

- if $T \dashv B$ a strong monad on \mathbb{V} , get $\mathcal{E}(T)$
with objects T -algebras and maps $f: A \xrightarrow{\Gamma} B$

Maps $f: \Gamma \times A \rightarrow B$ in \mathbb{V} that are right-linear

SEMANTICS

(values trivially
indexed over themselves) $\xleftrightarrow{+}$ (computations indexed
over values)

SEMANTICS

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

DEFN: A CBPV model consists of

- a cartesian category \mathcal{V} for the values

SEMANTICS

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

DEFN: A CBPV model consists of

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

SEMANTICS

(values trivially indexed over themselves) \rightleftarrows^{\pm} (computations indexed over values)

DEFⁿ: A CBPV model consists of

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

$$\text{self } \mathbb{V} \left\langle \begin{array}{c} \xrightarrow{F} \\ \dashv \\ \xleftarrow{u} \end{array} \right\rangle \mathbb{C}$$

$$\mathbb{V}(F_{xx}, u_B) \cong \mathbb{C}_F(Fx, \underline{B})$$

SEMANTICS

(values trivially indexed over themselves) \rightleftarrows^{\pm} (computations indexed over values)

DEFN: A CBPV model consists of

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

$$\text{self } \mathbb{V} \begin{array}{c} \xrightarrow{F} \\ \dashv \\ \xleftarrow{u} \end{array} \mathbb{C}$$

}

$$\mathbb{V}(F_{xx}, u_B) \cong \mathbb{C}_F(Fx, \underline{B})$$

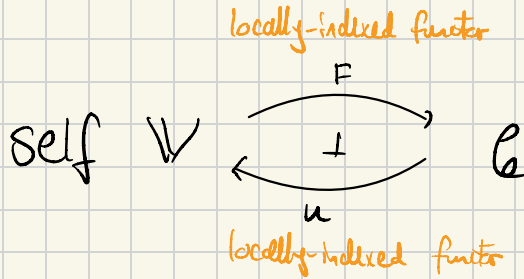
+ structure $f_a \rightarrow, +, \times$ etc.

SEMANTICS

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

VALUE TYPES $[A] \in \mathcal{V}$

COMPUTATION TYPES $[B] \in \mathcal{C}$



unit + counit give
rise to think / force

Semantics of CRPV [Levy]

Need to interpret

$\Gamma \Vdash V : A$
values

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

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

Need hints depending
on $\Gamma \in \text{Val}$ and $\underline{B}, \underline{C} \in \text{Comp}$

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

cf. $\llbracket M \rrbracket$ for a λ -term M
and context $\llbracket _ \rrbracket$

Semantics of CRPV [Levy]

Need to interpret

$$\Gamma \Vdash V : A$$

values

↓
in a cartesian
category \mathcal{V} , ie.
in $\mathcal{V}(\llbracket \Gamma \rrbracket, \llbracket A \rrbracket)$

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

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

computations

Need horns depending
on $\Gamma \in \text{Val}$ and $\underline{B}, \underline{C} \in \text{Comp}$

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

stacks / contexts

cf. $\llbracket \llbracket m \rrbracket \rrbracket$ for a λ -term m
and context $\llbracket _ \rrbracket$

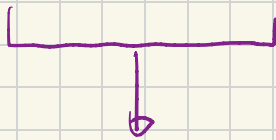
↓
in a locally \mathcal{V} -indexed
category \mathcal{C} , ie. in
 $\mathcal{C}_{\text{arr}}(\llbracket \underline{B} \rrbracket, \llbracket \underline{C} \rrbracket)$

Semantics of CRPV [Levy]

Need to interpret

$$\Gamma \Vdash V : A$$

values



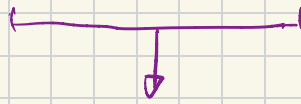
in a cartesian category \mathcal{V} , ie.
in $\mathcal{V}(\llbracket \Gamma \rrbracket, \llbracket A \rrbracket)$

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



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

computations



$$\text{in } \mathcal{V}(\llbracket \Gamma \rrbracket, \llbracket \underline{B} \rrbracket) \\ \cong \mathcal{C}_{\text{horn}}(F1, \llbracket \underline{B} \rrbracket)$$

Need horns depending
on $\Gamma \in \text{Val}$ and $\underline{B}, \underline{C} \in \text{Comp}$



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

stacks / contexts

cf. $\llbracket \llbracket m \rrbracket \rrbracket$ for a λ -term m
and context $\llbracket - \rrbracket$



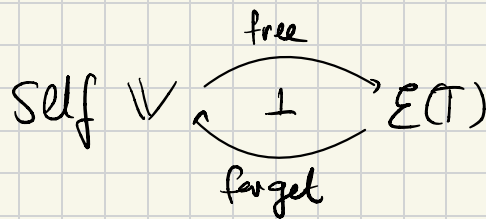
in a locally \mathcal{V} -indexed
category \mathcal{C} , ie. in
 $\mathcal{C}_{\text{horn}}(\llbracket \underline{B} \rrbracket, \llbracket \underline{C} \rrbracket)$

EXAMPLES:

many more in the
CBPU book!

1) ALGEBRA MODELS

for T a strong monad
on \mathcal{V}

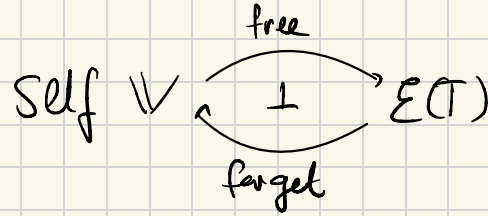


EXAMPLES:

many more in the
CBPU book!

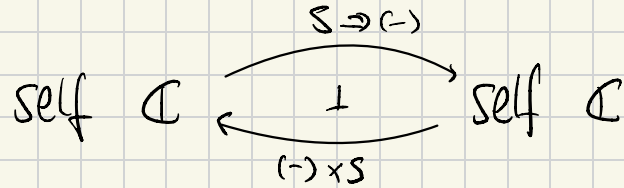
1) ALGEBRA MODELS

for T a strong monad
on \mathcal{V}



2) STATE MODELS

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

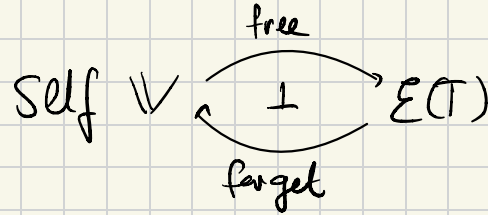


EXAMPLES:

many more in the
CBPU book!

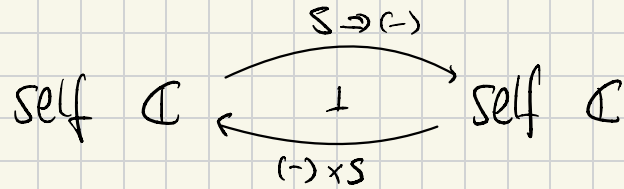
1) ALGEBRA MODELS

for T a strong monad
on \mathcal{V}



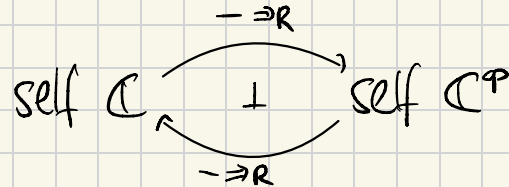
2) STATE MODELS

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



3) CONTINUATION MODELS

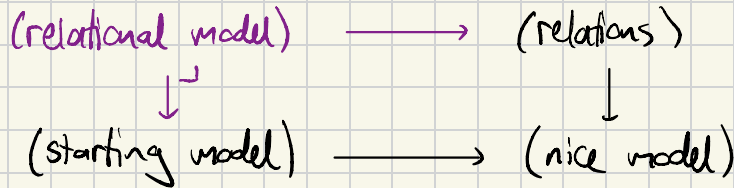
for a ccc \mathcal{C} and $R \in \mathcal{C}$



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



3) recover sensible syntactic defⁿ

$\{R_A \mid A \in \text{ValType}\}$
 $\{R_B \mid B \in \text{CompType}\}$

Qn: what is a CBPV fibration?

Prior work in the
algebraic case, see
eg- Kanamori, McDermott

Qn: what is a CBPV fibration?

Prior work in the
algebraic case, see
eg. Kanamori, 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
algebraic case, see
eg. Kanamori, 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?

Qn: what is a CBPV fibration?

A recipe:

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

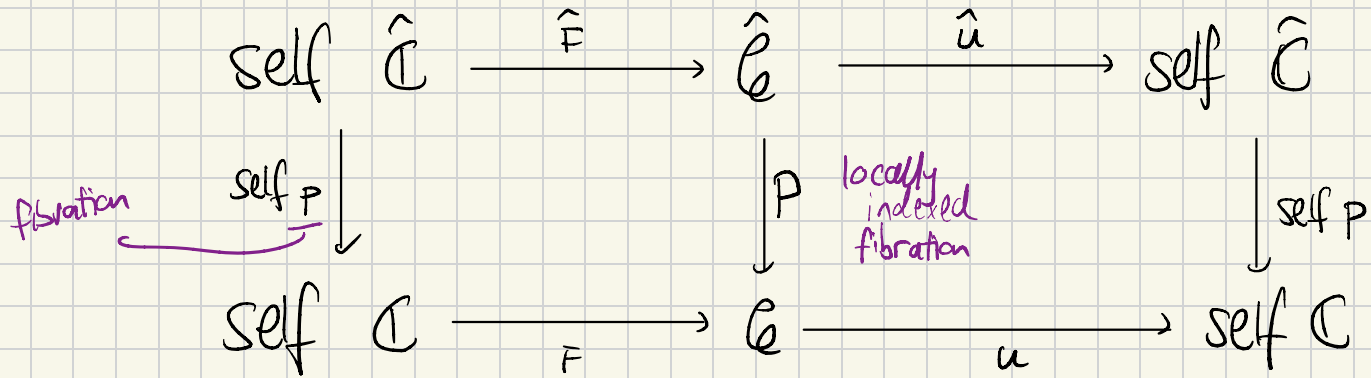
$$\text{self } \mathbb{C} \xrightarrow{F} \mathbb{C} \xrightarrow{u} \text{self } \mathbb{C}$$

CBPV fibrations

$$\text{self } \hat{\mathbb{C}} \xrightarrow{\hat{F}} \hat{\mathbb{C}} \xrightarrow{\hat{u}} \text{self } \hat{\mathbb{C}}$$

$$\text{self } \mathbb{C} \xrightarrow{F} \mathbb{C} \xrightarrow{u} \text{self } \mathbb{C}$$

CBPV fibrations



in the 2-category of locally-indexed categories

CBPV fibrations

$$\begin{array}{ccccc} \text{self } \hat{\mathbb{C}} & \xrightarrow{\hat{F}} & \hat{\mathbb{C}} & \xrightarrow{\hat{u}} & \text{self } \hat{\mathbb{C}} \\ \text{self } P \downarrow & & \downarrow P & & \downarrow \text{self } P \\ \text{self } \mathbb{C} & \xrightarrow{F} & \mathbb{C} & \xrightarrow{u} & \text{self } \mathbb{C} \end{array}$$

fibration (pointing to the left vertical arrow)

locally indexed (pointing to the middle vertical arrow)

... st the adjunction structure is preserved

from the general theory, we get:

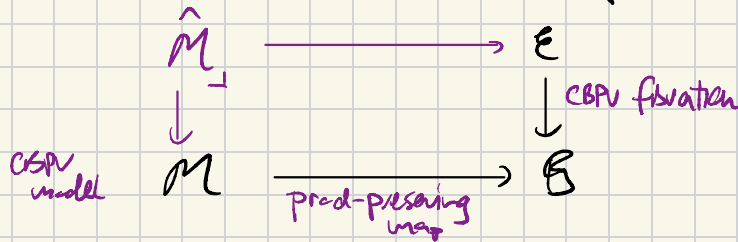
= some nice 2-category theory

from the general theory, we get:

1) nice closure properties

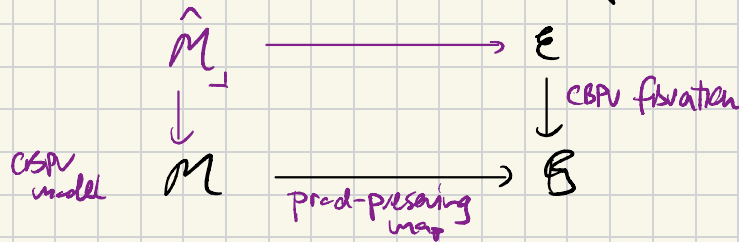
from the general theory, we get:

- 1) nice closure properties
- 2) universal constructions of relational models:



from the general theory, we get:

- 1) nice closure properties
- 2) universal constructions of relational models:

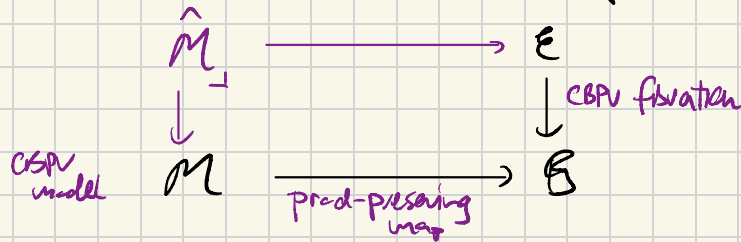


- 3) locally-indexed versions of key fibrations
for new-from-old (eg. subobject, codomain)

+ recovers the syntactic defⁿ of McDermott

from the general theory, we get:

- 1) nice closure properties
- 2) universal constructions of relational models:



* 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 defⁿ of McDermott

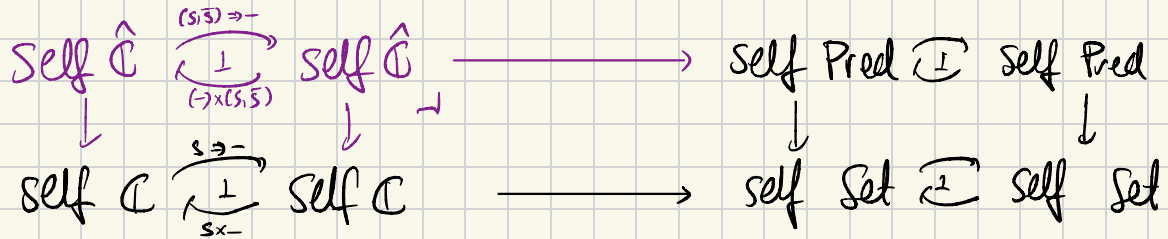
~ so essentially the theory we had for STC / CBV*

EXAMPLES.

i) algebra models : coincides with monad lifting

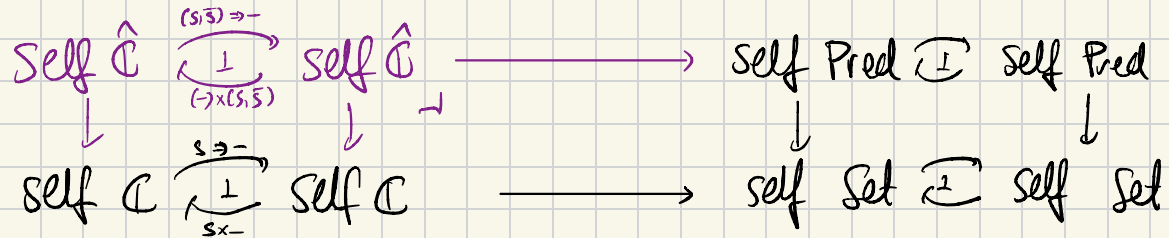
EXAMPLES.

- 1) algebra models : coincides with monad lifting
- 2) state / continuation models as expected :



EXAMPLES.

- 1) algebra models : coincides with monad lifting
- 2) state / continuation models as expected :



~ get logical relations as families of relations

$$\{R_A \mid A \in \text{ValType}\}, \quad \{R_B \mid B \in \text{CompType}\}$$

EFFECT SIMULATION

for algebra models for L and Spin
on Set

algebras
= monads

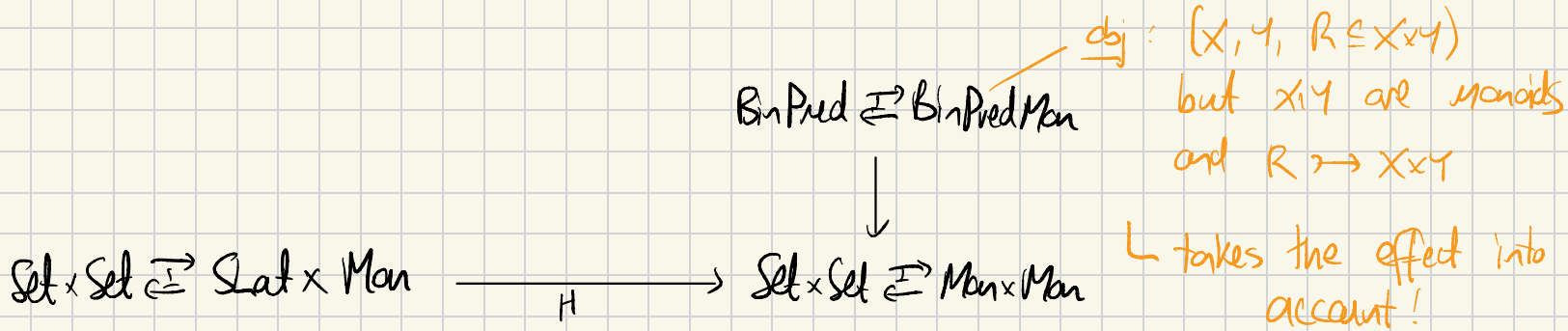
algebras
= sub-semilattices

EFFECT SIMULATION

for algebra models for L and Spin
on Set

algebras = monoids

algebras = sub-semilattices



EFFECT SIMULATION

for algebra models for L and Spin
 on Set

$\underbrace{\hspace{10em}}_{\text{algebras} = \text{monoids}}$
 $\underbrace{\hspace{10em}}_{\text{algebras} = \text{sub-semilattices}}$

obj: $(X, M, R \subseteq M \times M)$
 X : Set
 M : monoid
 R : submonoid

$\text{BinPred} \xrightarrow{\hat{F}} \hat{C}$
 \downarrow

$\text{Set} \times \text{Set} \xrightarrow{\hat{C}} \text{Set} \times \text{Mon}$

\longrightarrow

$\text{BinPred} \xrightarrow{\hat{C}} \text{BinPredMon}$

\downarrow

$\text{Set} \times \text{Set} \xrightarrow{\hat{C}} \text{Mon} \times \text{Mon}$

\xrightarrow{H}

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

but X, γ are monoids
 and $R \rightarrow X \times X$

L takes the effect into account!

EFFECT SIMULATION

for algebra models for L and Spin
on Set

algebras = monoids
algebras = sub-semilattices

obj: $(X, M, R \subseteq M \times M)$
 X : set
 M : monoid
 R : submonoid

$\text{BinPred} \xrightarrow{\hat{F}} \hat{C}$

$\text{Set} \times \text{Set} \xrightarrow{\cong} \text{Set} \times \text{Mon}$

$\text{BinPred} \cong \text{BinPredMon}$

\downarrow

$\text{Set} \times \text{Set} \cong \text{Mon} \times \text{Mon}$

\xrightarrow{H}

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

but X, γ are monoids
and $R \rightarrow X \times X$

L takes the effect into account!

$\hat{F}(A, B, R) = (P_{\hat{F}}A, LB, \hat{F}R)$ where $(p, e) \in \hat{F}R$ 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 \approx fibrations that preserve the model

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

Summing up:

- logical relations, denotationally \approx fibrations that preserve the model

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

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

Summing up:

- logical relations, denotationally \approx fibrations that preserve the model

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

- for CBU 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 STC / CBV.