Logical relations for CBPV models, via internal fibrations in a 2-category

LICS 2025, Singapore

These slides available at philipsaville.co.uk

Pedro H. Azevedo de Amorim (University of Oxford, UK)
Satoshi Kura (Waseda University, Japan)
Philip Saville (University of Sussex, UK)

What’s it all about?

What’s it all about?

the first complete denotational
account of logical relations for CBPV

What’s it all about?

the first complete denotational account of logical relations for CBPV

2025 Alonzo Church prize winner!

Levy’s call-by-push-value

= an effectful language with fine control over when effects happen

subsumes both CBV and CBN

What's it all about?

the first complete denotational account of logical relations for CBPV

a classic proof technique for logics and programming languages

"operational": termination, normalisation, …

"denotational": definability, simulation, …

= characterising the definable morphisms in a model

= how is $[\![P]\!]^{\mathscr{M}}$ related to $[\![P]\!]^{\mathscr{N}}$?

2025 Alonzo Church prize winner!

Levy's call-by-push-value

= an effectful language with fine control over when effects happen

subsumes both CBV and CBN

What's it all about?

logical relations as a way to build denotational models

related work:
Kammar (2014), McDermott (2020), …

other approaches:
Levy (2012), Goncharov—Tsampas—Urbat (2025), …

the first complete denotational account of logical relations for CBPV

a classic proof technique for logics and programming languages

"operational": termination, normalisation, …

"denotational": definability, simulation, …

= characterising the definable morphisms in a model

= how is $[\![P]\!]^{\mathscr{M}}$ related to $[\![P]\!]^{\mathscr{N}}$?

2025
Alonzo Church
prize winner!

Levy's call-by-push-value

= an effectful language with fine control over when effects happen

subsumes both CBV and CBN

What you get

1. Principled definition of relations models for CBPV
2. Lots of applications and examples, via a lifting theorem
   eg. effect simulation, $\top\top$-lifting, syntactic definition, Lafont argument, …
3. General theory encompassing STLC, λ_{ml} and CBPV

simply-typed λ-calculus

monadic metalanguage

Logical relations (in their simplest form) Milner, Plotkin, ….

A logical relation R consists of:

a predicate $R_A \subseteq [\![A]\!]$ for each type A…

…determined inductively at higher types by a logical relations condition

Logical relations (in their simplest form) Milner, Plotkin, ….

A logical relation R consists of:

> a predicate $R_A \subseteq [\![A]\!]$ for each type A…
>
> …determined inductively at higher types by a logical relations condition

Eg. logical relations condition for $\to$ in STLC:

$$f \in R_{A \to B} \subseteq [\![A]\!] \Rightarrow [\![B]\!] \qquad \text{iff} \qquad f(x) \in R_B \text{ whenever } x \in R_A$$

Logical relations (in their simplest form) Milner, Plotkin, ….

A logical relation R consists of:

> a predicate $R_A \subseteq [\![A]\!]$ for each type A…
>
> …determined inductively at higher types by a logical relations condition

Basic Lemma:

every program is in the relation: $[\![P]\!] \in R_A$ for any closed $P : A$

so long as this holds for the basic constants

Logical relations, denotationally

Statman,
Reynolds & Ma, ...

Logical relations, denotationally

Statman,
Reynolds & Ma, ….

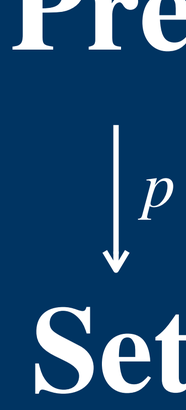
semantic model

Set

Logical relations, denotationally

Statman,
Reynolds & Ma, ….

objects: $(X \in \mathbf{Set}, R \subseteq X)$
maps: functions preserving the predicate



semantic model

Logical relations, denotationally

Statman,
Reynolds & Ma, ….

objects: $(X \in \mathbf{Set}, R \subseteq X)$
maps: functions preserving the predicate

semantic model

Pred is cartesian closed:

$$(X, R) \Rightarrow (Y, S) := (X \Rightarrow Y, R \supset S)$$

$f \in R \supset S$ iff $f(x) \in S$ whenever $x \in R$

$f \in [\![A \to B]\!]^{\mathbf{Pred}}$ iff f preserves the predicate on $[\![A]\!]^{\mathbf{Set}}$

Logical relations, denotationally

Statman,
Reynolds & Ma, ….

objects: $(X \in \mathbf{Set}, R \subseteq X)$
maps: functions preserving the predicate

$\mathbf{Pred}$

$\downarrow p$

$\mathbf{Set}$

semantic model

Pred is cartesian closed:

$$(X, R) \Rightarrow (Y, S) := (X \Rightarrow Y, R \supset S)$$

$f \in R \supset S$ iff $f(x) \in S$ whenever $x \in R$

$f \in [\![A \to B]\!]^{\mathbf{Pred}}$ iff f preserves the predicate on $[\![A]\!]^{\mathbf{Set}}$

CC-structure encodes the
logical relations condition!

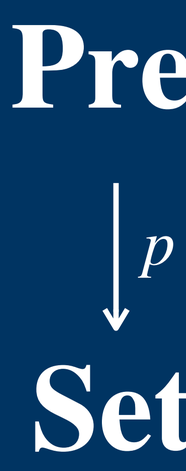
Logical relations, denotationally

Statman,
Reynolds & Ma, ….

objects: $(X \in \mathbf{Set}, R \subseteq X)$
maps: functions preserving the predicate

forgetful functor p
strictly preserves CC-structure

semantic model



Pred is cartesian closed:

$$(X, R) \Rightarrow (Y, S) := (X \Rightarrow Y, R \supset S)$$

$f \in R \supset S$ iff $f(x) \in S$ whenever $x \in R$

$f \in [\![A \to B]\!]^{\mathbf{Pred}}$ iff f preserves the predicate on $[\![A]\!]^{\mathbf{Set}}$

CC-structure encodes the
logical relations condition!

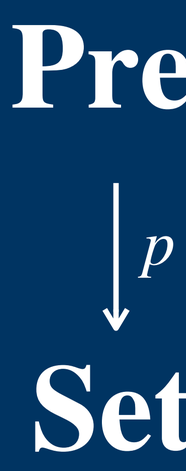
Logical relations, denotationally

Statman,
Reynolds & Ma, ….

objects: $(X \in \mathbf{Set}, R \subseteq X)$
maps: functions preserving the predicate

forgetful functor p
strictly preserves CC-structure

semantic model



Pred is cartesian closed:

$$(X, R) \Rightarrow (Y, S) := (X \Rightarrow Y, R \supset S)$$

$f \in R \supset S$ iff $f(x) \in S$ whenever $x \in R$

$f \in [\![A \to B]\!]^{\mathbf{Pred}}$ iff f preserves the predicate on $[\![A]\!]^{\mathbf{Set}}$

Basic Lemma holds automatically:

$$p\left([\![P]\!]^{\mathbf{Pred}}\right) = [\![P]\!]^{\mathbf{Set}}$$

CC-structure encodes the
logical relations condition!

Logical relations = relations models

Hermida, Jacobs, Katsumata, ….

Logical relations = relations models

Hermida, Jacobs, Katsumata, ….

semantic model $\mathscr{M}$

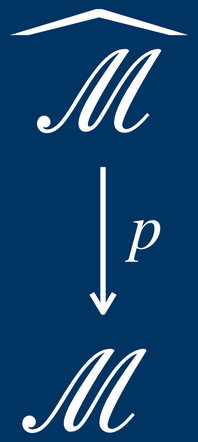
Logical relations = relations models

Hermida, Jacobs, Katsumata, ….

‘relations model’

forgetful functor p
strictly preserves model-structure

semantic model



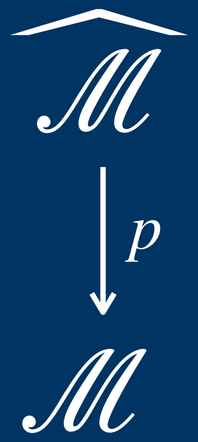
Logical relations = relations models

Hermida, Jacobs, Katsumata, ….

‘relations model’

forgetful functor p
strictly preserves model-structure

semantic model



model-structure encodes the
logical relations condition!

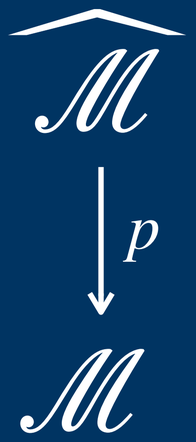
Logical relations = relations models

Hermida, Jacobs, Katsumata, ….

‘relations model’

forgetful functor p
strictly preserves model-structure

semantic model



model-structure encodes the
logical relations condition!

Basic Lemma holds automatically:

$$p\left([\![P]\!]^{\widehat{\mathscr{M}}}\right) = [\![P]\!]^{\mathscr{M}}$$

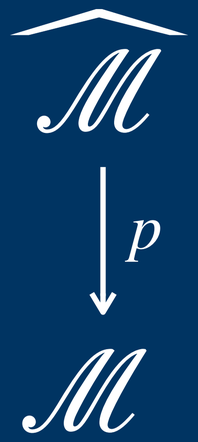
Logical relations = relations models

Hermida, Jacobs, Katsumata, ….

‘relations model’

forgetful functor p
strictly preserves model-structure

semantic model



model-structure encodes the
logical relations condition!

‘relations’ = p is a fibration

Basic Lemma holds automatically:

$$p\left([\![P]\!]^{\widehat{\mathscr{M}}}\right) = [\![P]\!]^{\mathscr{M}}$$

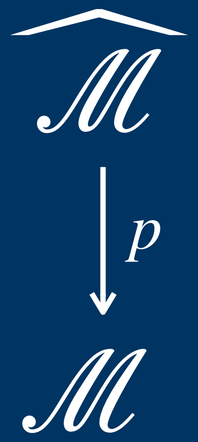
Logical relations = relations models

Hermida, Jacobs, Katsumata, ….

‘relations model’

forgetful functor p
strictly preserves model-structure

semantic model



model-structure encodes the
logical relations condition!

‘relations’ = p is a fibration

Basic Lemma holds automatically:

$$p\left([\![P]\!]^{\widehat{\mathscr{M}}}\right) = [\![P]\!]^{\mathscr{M}}$$

fibration for logical relations:
fibration
strictly preserving
model structure

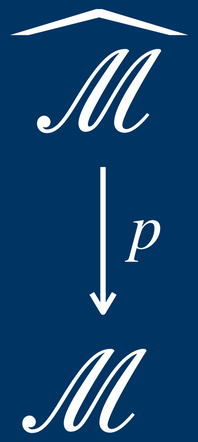
Logical relations = fibrations for logical relations

Hermida, Jacobs, Katsumata, ….

‘relations model’

forgetful functor p
strictly preserves model-structure

semantic model



model-structure encodes the
logical relations condition!

‘relations’ = p is a fibration

Basic Lemma holds automatically:

$$p\left([\![P]\!]^{\widehat{\mathscr{M}}}\right) = [\![P]\!]^{\mathscr{M}}$$

fibration for logical relations:
fibration
strictly preserving
model structure

Logical relations
$\cong$ relations models
$\cong$ fibrations for logical relations

fibration for logical relations:
fibration
strictly preserving
model structure

Building fibrations for logical relations

eg. effect simulation, $\top\top$-lifting, Kripke relations of varying arity, …

fibration for logical relations:

fibration
strictly preserving
model structure

Building fibrations for logical relations

eg. effect simulation, $\top\top$-lifting, Kripke relations of varying arity, …

fibration for logical relations:

fibration
strictly preserving
model structure

semantic
model

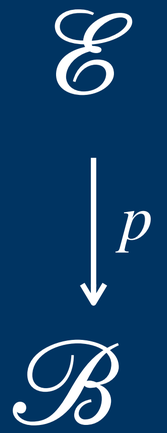
Building fibrations for logical relations

eg. effect simulation, TT-lifting, Kripke relations of varying arity, …

fibration for logical relations:

fibration
strictly preserving
model structure

semantic model $\mathscr{M}$



abstract notion of ‘relation’

ie. p a fibration for logical relations

Building fibrations for logical relations

eg. effect simulation, TT-lifting, Kripke relations of varying arity, …

fibration for logical relations:

fibration
strictly preserving
model structure

weak map of models

abstract notion of ‘relation’

ie. p a fibration for logical relations

Building fibrations for logical relations

eg. effect simulation, TT-lifting, Kripke relations of varying arity, …

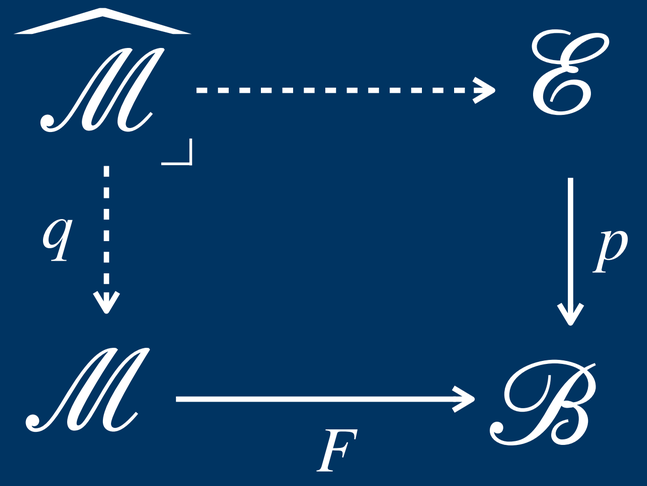
fibration for logical relations:

fibration
strictly preserving
model structure

lifting theorem:

universal choice of model

‘glueing’ objects of $\mathscr{M}$ to ‘relations’ in $\mathscr{E}$



semantic model

weak map of models

abstract notion of ‘relation’

ie. p a fibration for logical relations

Building fibrations for logical relations

eg. effect simulation, $\top\top$-lifting, Kripke relations of varying arity, …

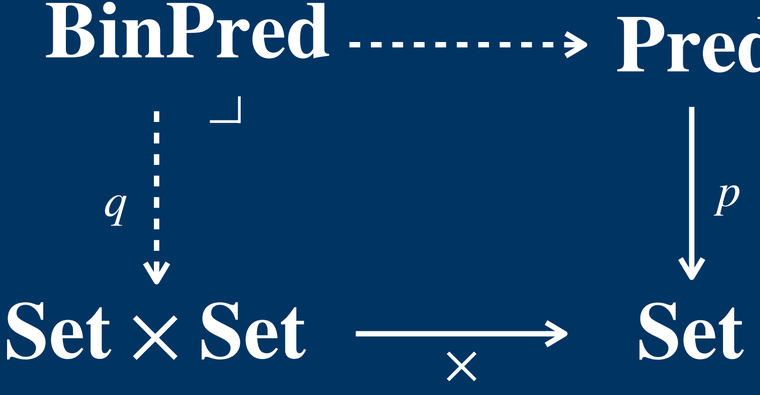
fibration for logical relations:

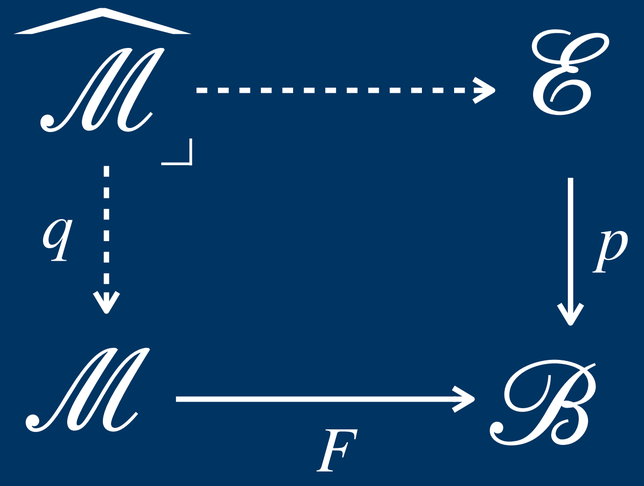
fibration
strictly preserving
model structure

lifting theorem:

universal choice of model
'glueing' objects of $\mathscr{M}$ to 'relations' in $\mathscr{E}$

objects: $(X \in \mathbf{Set}, R \subseteq X \times X)$
maps: pairs of maps preserving the relation





weak map of models

abstract notion of 'relation'

ie. p a fibration for logical relations

simply-typed
λ-calculus

monadic
metalanguage

This works for STLC and λ_{ml} and λ_{c}

computational
λ-calculus

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

simply-typed λ-calculus

monadic metalanguage

This works for STLC and λ_{ml} and λ_{c}

computational λ-calculus

…but not obvious for CBPV

simply-typed λ-calculus

monadic metalanguage

This works for STLC and λ_{ml} and λ_{c}

computational λ-calculus

…but not obvious for CBPV

Why?

semantic models are more subtle:
uses enriched categories

want to say ‘fibration preserving model structure’,
but what is the right notion of fibration?

What we do:

turn to 2-category theory

= a language for structures on categories

What we do:

turn to 2-category theory

= a language for structures on categories

Existing fibrations for logical relations

= fibrations internal to 2-categories of models

What we do:

turn to 2-category theory

= a language for structures on categories

Existing fibrations for logical relations
= fibrations internal to 2-categories of models

Construct a 2-category of CBPV models

→ fibration for logical relations := fibrations internal to this

→ key examples from the general theory eg. subobject, codomain, …

What we do:

turn to 2-category theory

= a language for structures on categories

Existing fibrations for logical relations
= fibrations internal to 2-categories of models

Construct a 2-category of CBPV models

→ fibration for logical relations := fibrations internal to this

→ key examples from the general theory eg. subobject, codomain, …

Prove a lifting theorem encompassing STLC, λ_{ml} and CBPV

What we do:

turn to 2-category theory

= a language for structures on categories

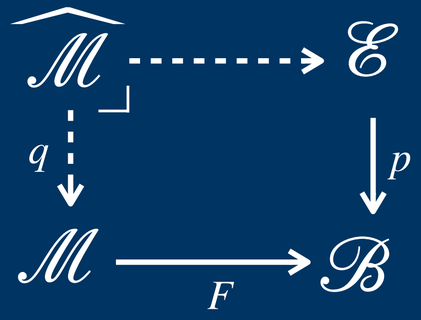
Existing fibrations for logical relations
= fibrations internal to 2-categories of models

Construct a 2-category of CBPV models

→ fibration for logical relations := fibrations internal to this

→ key examples from the general theory eg. subobject, codomain, …

Prove a lifting theorem encompassing STLC, λ_{ml} and CBPV



What we do:

turn to 2-category theory

= a language for structures on categories

Existing fibrations for logical relations
= fibrations internal to 2-categories of models

Construct a 2-category of CBPV models

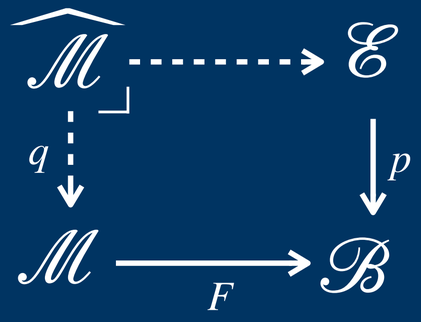
→ fibration for logical relations := fibrations internal to this

→ key examples from the general theory eg. subobject, codomain, …

Prove a lifting theorem encompassing STLC, λ_{ml} and CBPV

Get many new CBPV models

eg. effect simulation, ⊤⊤-lifting, syntactic defn, conservativity,…



A complete denotational account of logical relations for CBPV

A complete denotational account of logical relations for CBPV

1. Principled definition of relations models for CBPV
2. Lots of applications and examples, via a lifting theorem
   eg. effect simulation, TT-lifting, syntactic definition, Lafont argument, …
3. General theory encompassing STLC, λ_{ml} and CBPV

simply-typed λ-calculus

monadic metalanguage

A complete denotational account of logical relations for CBPV

1. Principled definition of relations models for CBPV
2. Lots of applications and examples, via a lifting theorem
   eg. effect simulation, $\top\top$-lifting, syntactic definition, Lafont argument, …
3. General theory encompassing STLC, λ_{ml} and CBPV

simply-typed λ-calculus

monadic metalanguage

email me: p.saville@sussex.ac.uk
arxiv link: 10.48550/arXiv.2505.14482
these slides available at philipsaville.co.uk