Fully abstract models for effectful λ-calculi via category-theoretic logical relations

Ohad Kammar[†] and Shin-ya Katsumata[*] and Philip Saville[‡]

[†]School of Informatics
University of Edinburgh

[*]National Institute of Informatics
Tokyo

[‡]Department of Computer Science
University of Oxford

preprint & these slides at philipsaville.co.uk

This work

with sum types

A category-theoretic construction that:

takes a [suitable] model of an effectful λ-calculus

. . . and returns an adequate & fully-abstract model

This work

with sum types

A category-theoretic construction that:

takes a [suitable] model of an effectful λ-calculus

. . . and returns an adequate & fully-abstract model

This work

with sum types

A category-theoretic construction that:

takes a [suitable] model of an effectful λ-calculus

... and returns an adequate & fully-abstract model

This work

with sum types

A category-theoretic construction that:

takes a [suitable] model of an effectful λ-calculus

. . . and returns an adequate & fully-abstract model

Adequacy and full abstraction

Contextual equivalence [Morris, Milner,...]

$$\Gamma \vdash M \simeq_{\text{ctx}} M' : \sigma \quad \Longleftrightarrow \quad \mathcal{C}[M] \Downarrow V \iff \mathcal{C}[M'] \Downarrow V$$

$\mathcal{C}[-]$ any closed ground context

Intuition:
swapping M and M'
doesn't affect
observable behaviour

Contextual equivalence [Morris, Milner, ...]

$$\Gamma \vdash M \simeq_{\mathrm{ctx}} M' : \sigma \iff \mathcal{C}[M] \Downarrow V \iff \mathcal{C}[M'] \Downarrow V$$

$\mathcal{C}[-]$ any closed ground context

Reasoning about $\simeq_{\mathrm{ctx}}$ is hard

$\rightsquigarrow$ motivates semantic interpretation $[\![M]\!]$

Contextual equivalence [Morris, Milner,. . .]

$$\Gamma \vdash M \simeq_{\mathrm{ctx}} M' : \sigma \iff \mathcal{C}[M] \Downarrow V \iff \mathcal{C}[M'] \Downarrow V$$

$\mathcal{C}[-]$ any closed ground context

Reasoning about $\simeq_{\mathrm{ctx}}$ is hard

$\rightsquigarrow$ motivates semantic interpretation $[\![M]\!]$

How does $[\![M]\!] = [\![M']\!]$ relate to $M \simeq_{\mathrm{ctx}} M'$?

[*c.f.* soundness and completeness in logic]

Contextual equivalence [Morris, Milner,...]

$$\Gamma \vdash M \simeq_{\mathrm{ctx}} M' : \sigma \iff \mathcal{C}[M] \Downarrow V \iff \mathcal{C}[M'] \Downarrow V$$

$\mathcal{C}[-]$ any closed ground context

Reasoning about $\simeq_{\mathrm{ctx}}$ is hard

$\rightsquigarrow$ motivates semantic interpretation $[\![M]\!]$

How does $[\![M]\!] = [\![M']\!]$ relate to $M \simeq_{\mathrm{ctx}} M'$?

[*c.f.* soundness and completeness in logic]

Adequacy: $[\![M]\!] = [\![M']\!] \implies M \simeq_{\mathrm{ctx}} M'$

Full abstraction: $M \simeq_{\mathrm{ctx}} M' \implies [\![M]\!] = [\![M']\!]$

Contextual equivalence [Morris, Milner,...]

$\Gamma \vdash M \simeq_{\mathrm{ctx}} M' : \sigma \iff \mathcal{C}[M] \Downarrow V \iff \mathcal{C}[M'] \Downarrow V$

$\mathcal{C}[-]$ any closed ground context

Reasoning about $\simeq_{\mathrm{ctx}}$ is hard

$\leadsto$ motivates semantic interpretation $[\![M]\!]$

How does $[\![M]\!] = [\![M']\!]$ relate to $M \simeq_{\mathrm{ctx}} M'$?

[*c.f.* soundness and completeness in logic]

Adequacy: $[\![M]\!] = [\![M']\!] \implies M \simeq_{\mathrm{ctx}} M'$

Full abstraction: $M \simeq_{\mathrm{ctx}} M' \implies [\![M]\!] = [\![M']\!]$

In an adequate, fully abstract model
semantic equality characterises contextual equivalence

Effectful λ-calculi

Effectful λ-calculi: syntax

Specified by a signature: you choose

Effectful λ-calculi: syntax

Specified by a signature: you choose

- A monadic effect *e.g.* exceptions

Effectful λ-calculi: syntax

Specified by a signature: you choose

- A monadic effect *e.g.* exceptions
- Base types $\mathrm{nat}, \mathrm{bool}, \ldots$

Effectful λ-calculi: syntax

Specified by a signature: you choose

- A monadic effect — *e.g.* exceptions
- Base types — $\mathrm{nat}, \mathrm{bool}, \ldots$
- Effectful operations — $\mathrm{raise}_{\mathrm{e}}, \ldots$

Effectful λ-calculi: syntax

Specified by a signature: you choose

- A monadic effect — *e.g.* exceptions
- Base types — $\mathrm{nat}, \mathrm{bool}, \ldots$
- Effectful operations — $\mathrm{raise}_{\mathrm{e}}, \ldots$
- Primitives — $\underline{\mathrm{n}} : \mathrm{nat}, \mathrm{and} : \mathrm{bool} * \mathrm{bool} \to \mathrm{bool}, \ldots$

Effectful λ-calculi: syntax

Specified by a signature: you choose

- A monadic effect — *e.g.* exceptions
- Base types — $\mathrm{nat}, \mathrm{bool}, \ldots$
- Effectful operations — $\mathrm{raise}_{\mathrm{e}}, \ldots$
- Primitives — $\underline{\mathrm{n}} : \mathrm{nat}, \mathrm{and} : \mathrm{bool} * \mathrm{bool} \to \mathrm{bool}, \ldots$

$\rightsquigarrow$ determines a HO language with products & sums

e.g. a HO language with exceptions

Effectful λ-calculi: signatures

- A monadic effect — *e.g.* exceptions
- Base types — $\mathrm{nat}, \mathrm{bool}, \ldots$
- Effectful operations — $\mathrm{raise}_{\mathrm{e}}, \ldots$
- Primitives — $\underline{\mathrm{n}} : \mathrm{nat}, \mathrm{and} : \mathrm{bool} * \mathrm{bool} \to \mathrm{bool}, \ldots$

Effectful λ-calculi: signatures

- A monadic effect — *e.g.* exceptions
- Base types — nat, bool, ...
- Effectful operations — $\mathrm{raise}_{\mathrm{e}}, \ldots$
- Primitives — $\underline{\mathrm{n}} : \mathrm{nat}, \mathrm{and} : \mathrm{bool} * \mathrm{bool} \to \mathrm{bool}, \ldots$

Effectful λ-calculi: semantics [à la Moggi]

Specified by a model: you choose

Effectful λ-calculi: signatures

- A monadic effect — *e.g.* exceptions
- Base types — nat, bool, ...
- Effectful operations — $\mathrm{raise}_{\mathrm{e}}, \ldots$
- Primitives — $\underline{\mathrm{n}} : \mathrm{nat}, \mathrm{and} : \mathrm{bool} * \mathrm{bool} \to \mathrm{bool}, \ldots$

Effectful λ-calculi: semantics [à la Moggi]

Specified by a model: you choose

- A CCC $\mathcal{M}$ with $(0, +)$ — *e.g.* Set

Effectful λ-calculi: signatures

- A monadic effect — *e.g.* exceptions
- Base types — $\mathrm{nat}, \mathrm{bool}, \ldots$
- Effectful operations — $\mathrm{raise}_{\mathrm{e}}, \ldots$
- Primitives — $\underline{\mathrm{n}} : \mathrm{nat}, \mathrm{and} : \mathrm{bool} * \mathrm{bool} \to \mathrm{bool}, \ldots$

Effectful λ-calculi: semantics [à la Moggi]

Specified by a model: you choose

- A CCC $\mathcal{M}$ with $(0, +)$ — *e.g.* Set
- Strong monad T — $T(X) = X + E$

Effectful λ-calculi: signatures

- A monadic effect — *e.g.* exceptions
- Base types — nat, bool, ...
- Effectful operations — $\mathrm{raise}_{\mathrm{e}}, \ldots$
- Primitives — $\underline{\mathrm{n}} : \mathrm{nat}, \mathrm{and} : \mathrm{bool} * \mathrm{bool} \to \mathrm{bool}, \ldots$

Effectful λ-calculi: semantics [à la Moggi]

Specified by a model: you choose

- A CCC $\mathcal{M}$ with $(0, +)$ — *e.g.* Set
- Strong monad T — $T(X) = X + E$
- $[\![\beta]\!] \in \mathcal{M}$ for each base type β — $[\![\mathrm{bool}]\!] = 2, [\![\mathrm{nat}]\!] = \mathbb{N}$

Effectful λ-calculi: signatures

- A monadic effect — *e.g.* exceptions
- Base types — $\mathrm{nat}, \mathrm{bool}, \ldots$
- Effectful operations — $\mathrm{raise_e}, \ldots$
- Primitives — $\underline{\mathrm{n}} : \mathrm{nat}, \mathrm{and} : \mathrm{bool} * \mathrm{bool} \to \mathrm{bool}, \ldots$

Effectful λ-calculi: semantics [à la Moggi]

Specified by a model: you choose

- A CCC $\mathcal{M}$ with $(0, +)$ — *e.g.* Set
- Strong monad T — $T(X) = X + E$
- $[\![\beta]\!] \in \mathcal{M}$ for each base type β — $[\![\mathrm{bool}]\!] = 2, [\![\mathrm{nat}]\!] = \mathbb{N}$
- Arrows interpreting the operations and primitives — $[\![\mathrm{raise_e}]\!] = \lambda x \,.\, \mathrm{inl}(e)$, $[\![\underline{\mathrm{n}}]\!] = (* \mapsto n : 1 \to \mathbb{N})$

Effectful λ-calculi: signatures

- A monadic effect — *e.g.* exceptions
- Base types — nat, bool, ...
- Effectful operations — $\mathrm{raise}_{\mathrm{e}}$, ...
- Primitives — $\underline{\mathrm{n}}$: nat, and : bool $*$ bool $\to$ bool, ...

Effectful λ-calculi: semantics [à la Moggi]

Specified by a model: you choose

- A CCC $\mathcal{M}$ with $(0, +)$ — *e.g.* Set
- Strong monad T — $T(X) = X + E$
- $[\![\beta]\!] \in \mathcal{M}$ for each base type β — $[\![\mathrm{bool}]\!] = 2$, $[\![\mathrm{nat}]\!] = \mathbb{N}$
- Arrows interpreting the operations and primitives — $[\![\mathrm{raise}_{\mathrm{e}}]\!] = \lambda x \,.\, \mathrm{inl}(e)$, $[\![\underline{\mathrm{n}}]\!] = (* \mapsto n : 1 \to \mathbb{N})$

$\rightsquigarrow$ determines an interpretation $[\![\Gamma \vdash M : \sigma]\!] : [\![\Gamma]\!] \to T[\![\sigma]\!]$

This work

signature

= chosen base types,
effect operations,
& primitives

+

semantic model

= CCC with coproducts $\mathcal{M}$
+ strong monad T
+ interpretation
+ conditions on $\mathcal{M}$, interp.

sufficient: full definability
at base types

$\Downarrow$

fully abstract model $\mathrm{OHR}(\mathcal{M})$
of computational λ-calculus + constants + sums

inspired by O'Hearn & Riecke's PCF model, 1995

concrete over $\mathcal{M}$:
maps in $\mathrm{OHR}(\mathcal{M})$ are
maps in $\mathcal{M}$ satisfying predicates

signature

= chosen base types,
effect operations,
& primitives

+

semantic model

= CCC with coproducts $\mathcal{M}$
+ strong monad T
+ interpretation
+ conditions on $\mathcal{M}$, interp.

sufficient: full definability
at base types

$\Downarrow$

fully abstract model $\mathrm{OHR}(\mathcal{M})$
of computational λ-calculus + constants + sums

inspired by O'Hearn & Riecke's PCF model, 1995

concrete over $\mathcal{M}$:
maps in $\mathrm{OHR}(\mathcal{M})$ are
maps in $\mathcal{M}$ satisfying predicates

signature

= chosen base types,
effect operations,
& primitives

+

semantic model

= CCC with coproducts $\mathcal{M}$
+ strong monad T
+ interpretation
+ conditions on $\mathcal{M}$, interp.

sufficient: full definability
at base types

$\Downarrow$

fully abstract model $\mathrm{OHR}(\mathcal{M})$
of computational λ-calculus + constants + sums

inspired by O'Hearn & Riecke's PCF model, 1995

concrete over $\mathcal{M}$:
maps in $\mathrm{OHR}(\mathcal{M})$ are
maps in $\mathcal{M}$ satisfying predicates

signature

= chosen base types,
effect operations,
& primitives

+

semantic model

= CCC with coproducts $\mathcal{M}$
+ strong monad T
+ interpretation
+ conditions on $\mathcal{M}$, interp.

sufficient: full definability
at base types

$\Downarrow$

fully abstract model $\mathrm{OHR}(\mathcal{M})$
of computational λ-calculus + constants + sums

inspired by O'Hearn & Riecke's PCF model, 1995

concrete over $\mathcal{M}$:
maps in $\mathrm{OHR}(\mathcal{M})$ are
maps in $\mathcal{M}$ satisfying predicates

signature

= chosen base types,
effect operations,
& primitives

+

semantic model

= CCC with coproducts $\mathcal{M}$
+ strong monad T
+ interpretation
+ conditions on $\mathcal{M}$, interp.

$\Downarrow$

sufficient: full definability
at base types

fully abstract model $\mathrm{OHR}(\mathcal{M})$
of computational λ-calculus + constants + sums

inspired by O'Hearn & Riecke's PCF model, 1995

concrete over $\mathcal{M}$:
maps in $\mathrm{OHR}(\mathcal{M})$ are
maps in $\mathcal{M}$ satisfying predicates

Cranking the handle

signature

e.g. base types $\mathrm{nat}, \mathsf{bool}$
+ primitives $\mathrm{tt}, \mathrm{ff}, \underline{\mathrm{n}}$ for $n \in \mathbb{N}$
+ effect operation $\mathrm{read}, \ldots$

$+$

semantic model

e.g. subcategory Set_κ of Set
+ reader monad R
+ $[\![\mathrm{nat}]\!] = \mathbb{N}$,
$[\![\mathsf{bool}]\!] = \{0, 1\}, \ldots$

not fully abstract!
(Matache & Staton)

$\Downarrow$

fully abstract model
of read-only state

Cranking the handle

signature

e.g. base type real

+ primitive $\underline{f}$ for each measurable f

+ effect operations

$\mathrm{sample}, \mathrm{score}, \mathrm{normalise}, \ldots$

$+$

semantic model

e.g. small sub-CCC of Qbs

+ probability monad

+ $[\![\mathrm{real}]\!] = (\mathbb{R}, \Sigma_{\mathbb{R}})$

$\Downarrow$

fully abstract model
of idealised probabilistic programming language

The OHR construction

Obstruction to full abstraction:

∃ 'bad' morphisms expressing behaviour the syntax cannot

[*c.f.* parallel-or]

The big picture

Obstruction to full abstraction:

$\exists$ 'bad' morphisms expressing behaviour the syntax cannot

[*c.f.* parallel-or]

$M \simeq_{\mathrm{ctx}} M' \quad \Longrightarrow \quad [\![M]\!](i) = [\![M']\!](i)$

for all 'program-like' inputs i

The big picture

Obstruction to full abstraction:

$\exists$ 'bad' morphisms expressing behaviour the syntax cannot

[*c.f.* parallel-or]

$M \simeq_{\mathrm{ctx}} M'$ $\implies$ $[\![M]\!](i) = [\![M']\!](i)$
for all 'program-like' inputs i

κ bad

The big picture

Obstruction to full abstraction:

$\exists$ 'bad' morphisms expressing behaviour the syntax cannot

[*c.f.* parallel-or]

$M \simeq_{\mathrm{ctx}} M'$	$\Longrightarrow$	$[\![M]\!](i) = [\![M']\!](i)$ for all 'program-like' inputs i
κ bad	$\Longrightarrow$	can have $[\![M]\!](\kappa) \neq [\![M']\!](\kappa)$

The big picture

Obstruction to full abstraction:

$\exists$ 'bad' morphisms expressing behaviour the syntax cannot

[*c.f.* parallel-or]

$M \simeq_{\mathrm{ctx}} M'$	$\implies$	$[\![M]\!](i) = [\![M']\!](i)$ for all 'program-like' inputs i
κ bad	$\implies$	can have $[\![M]\!](\kappa) \neq [\![M']\!](\kappa)$

Solution:

refine the model to remove all bad morphisms

The big picture

Obstruction to full abstraction:

$\exists$ 'bad' morphisms expressing behaviour the syntax cannot

[*c.f.* parallel-or]

Solution:

refine the model to remove all bad morphisms

What follows:

1. A general construction for refining models

   [hom-sets and function spaces!]

2. How to instantiate to remove all bad morphisms

A general construction for refining models

Aim: refine a category $\mathcal{M}$ so maps all satisfy certain properties

Aim: refine a category $\mathcal{M}$ so maps all satisfy certain properties

---

Example: category Pred over Set:

objects: pairs $(W \in \mathrm{Set}, \text{predicate } A \subseteq W)$

maps: maps in $\mathcal{M}$ preserving the predicates

Aim: refine a category $\mathcal{M}$ so maps all satisfy certain properties

---

The category $\mathrm{Pred}(\mathcal{M})$:

objects: pairs ($W \in \mathcal{M}$, 'relation' on W)

[unary, *n*-ary, varying arity; families of relations,. . .]

maps: maps in $\mathcal{M}$ preserving the relations

Aim: refine a category $\mathcal{M}$ so maps all satisfy certain properties

---

The category $\mathrm{Pred}(\mathcal{M})$:

objects: pairs ($W \in \mathcal{M}$, 'relation' on W)

[unary, *n*-ary, varying arity; families of relations,. . .]

maps: maps in $\mathcal{M}$ preserving the relations

$\rightsquigarrow$ internalise to function spaces: restrict to "concrete" relations

Aim: refine a category $\mathcal{M}$ so maps all satisfy certain properties

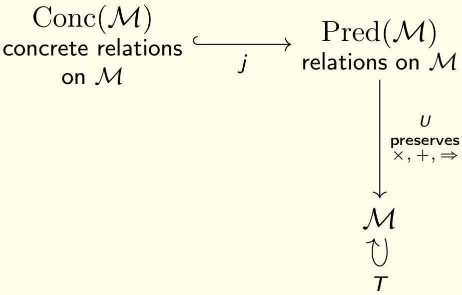
---

The category $\mathrm{Pred}(\mathcal{M})$:

objects: pairs $(W \in \mathcal{M}, \text{‘relation’ on } W)$
[unary, n-ary, varying arity; families of relations, . . .]

maps: maps in $\mathcal{M}$ preserving the relations

⇝ internalise to function spaces: restrict to “concrete” relations



Aim: refine a category $\mathcal{M}$ so maps all satisfy certain properties

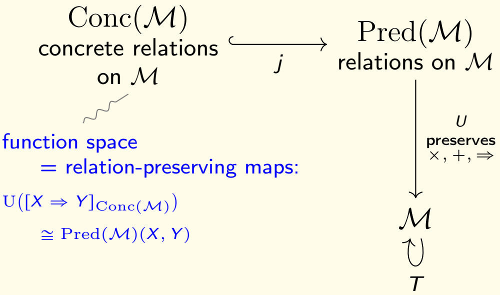
The category $\mathrm{Pred}(\mathcal{M})$:

objects: pairs ($W \in \mathcal{M}$, 'relation' on W)

[unary, n-ary, varying arity; families of relations,. . .]

maps: maps in $\mathcal{M}$ preserving the relations

⇝ internalise to function spaces: restrict to "concrete" relations



Aim: refine a category $\mathcal{M}$ so maps all satisfy certain properties

---

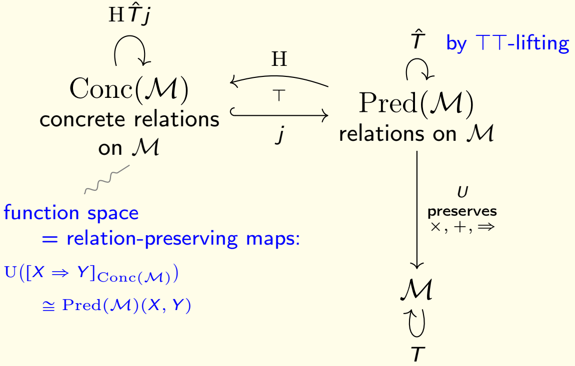
The category $\mathrm{Pred}(\mathcal{M})$:

objects: pairs $(W \in \mathcal{M}$, 'relation' on $W)$

[unary, n-ary, varying arity; families of relations, . . .]

maps: maps in $\mathcal{M}$ preserving the relations

$\rightsquigarrow$ internalise to function spaces: restrict to "concrete" relations



Aim: refine a category $\mathcal{M}$ so maps all satisfy certain properties

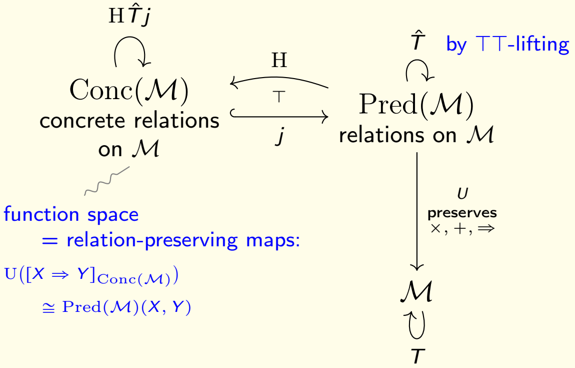
The category $\mathrm{Pred}(\mathcal{M})$:

objects: pairs $(W \in \mathcal{M},$ 'relation' on $W)$

[unary, n-ary, varying arity; families of relations, . . .]

maps: maps in $\mathcal{M}$ preserving the relations

$\rightsquigarrow$ internalise to function spaces: restrict to "concrete" relations



Aim: refine a category $\mathcal{M}$ so maps all satisfy certain properties

---

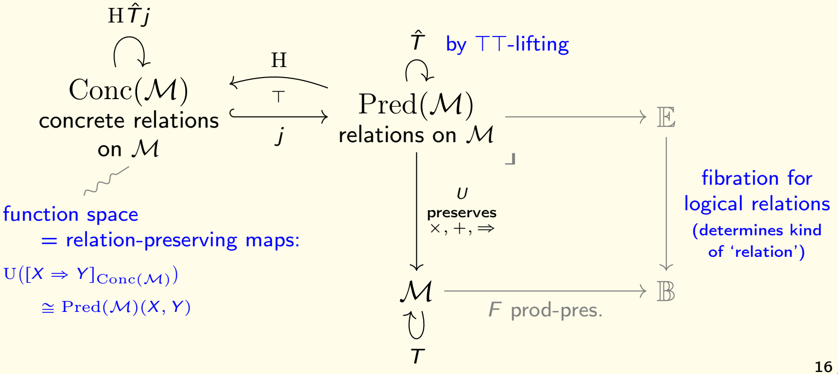
The category $\mathrm{Pred}(\mathcal{M})$:

objects: pairs $(W \in \mathcal{M}, \text{'relation' on } W)$

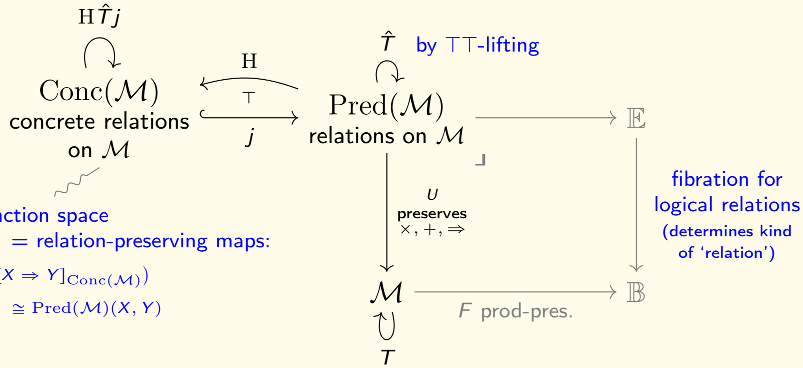
[unary, *n*-ary, varying arity; families of relations, . . .]

maps: maps in $\mathcal{M}$ preserving the relations

$\rightsquigarrow$ internalise to function spaces: restrict to "concrete" relations



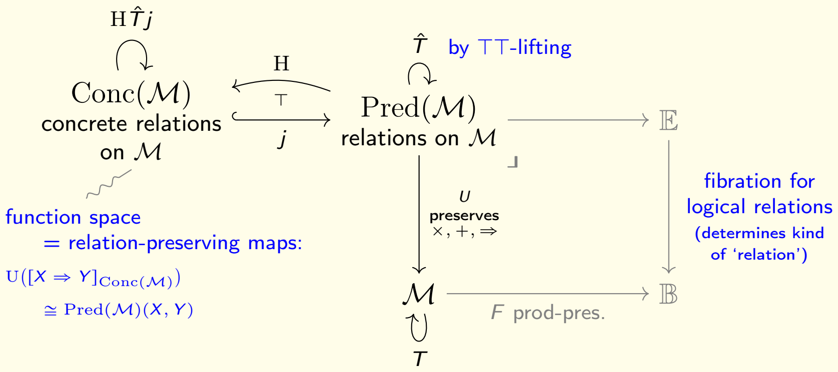
A general construction for refining a model $\mathcal{M}$



function space
= relation-preserving maps:

$\mathrm{U}([X \Rightarrow Y]_{\mathrm{Conc}(\mathcal{M})})$
$\cong \mathrm{Pred}(\mathcal{M})(X, Y)$

A general construction for refining a model $\mathcal{M}$



$\rightsquigarrow$ $\mathrm{OHR}(\mathcal{M})$ will be $\mathrm{Conc}(\mathcal{M})$ for a careful choice of "relations"

The big picture

Obstruction to full abstraction:

$\exists$ 'bad' morphisms expressing behaviour the syntax cannot

[*c.f.* parallel-or]

Solution:

refine the model to remove all bad morphisms

What follows:

1. A general construction for refining models

   [hom-sets and function spaces!]

2. How to instantiate to remove all bad morphisms

The big picture

Obstruction to full abstraction:

$\exists$ 'bad' morphisms expressing behaviour the syntax cannot

[*c.f.* parallel-or]

Solution:

refine the model to remove all bad morphisms

What follows:

✓ A general construction for refining models

[hom-sets and function spaces!]

? How to instantiate to remove all bad morphisms

Instantiating the general construction

If f is definable ($f = [\![M]\!]$) . . .

If f is definable $(f = [\![M]\!])$ … it can't be bad

If f is definable ($f = [\![M]\!]$) ... it can't be bad

suggests: suffices to cut out all non-definable maps

If f is definable ($f = [\![M]\!]$) … it can't be bad

suggests: suffices to cut out all non-definable maps

Lemma: [*c.f.* Curien's "definable separability condition"]

any well-pointed model in which every map $[\![\Gamma]\!] \to T[\![\sigma]\!]$ is definable is fully abstract.

$f = g : X \to Y$
iff
$f \circ \gamma = g \circ \gamma$ for all $\gamma : 1 \to X$

If f is definable ($f = [\![M]\!]$) ... it can't be bad

suggests: suffices to cut out all non-definable maps

Lemma: [*c.f.* Curien's "definable separability condition"]

any well-pointed model in which every map $[\![\Gamma]\!] \to T[\![\sigma]\!]$ is definable is fully abstract.

$f = g : X \to Y$
iff
$f \circ \gamma = g \circ \gamma$ for all $\gamma : 1 \to X$

Question: which relations guarantee definability?

[Plotkin, Jung & Tiuryn, Alimohamed, ...]

If f is definable ($f = [\![M]\!]$) ... it can't be bad

suggests: suffices to cut out all non-definable maps

Lemma: [*c.f.* Curien's "definable separability condition"]

any well-pointed model in which every map $[\![\Gamma]\!] \to T[\![\sigma]\!]$ is definable is fully abstract.

$f = g : X \to Y$
iff
$f \circ \gamma = g \circ \gamma$ for all $\gamma : 1 \to X$

Question: which relations guarantee definability?

[Plotkin, Jung & Tiuryn, Alimohamed, ...]

---

f is definable $\iff$ f preserves every logical relation

type-indexed family of relations
compatible with type- & term-formers

Question: which relations guarantee definability?

[Plotkin, Jung & Tiuryn, Alimohamed, . . .]

f is definable $\iff$ f preserves every logical relation

type-indexed family of relations
compatible with type- & term-formers

Question: which relations guarantee definability?

[Plotkin, Jung & Tiuryn, Alimohamed, . . .]

---

f is definable $\iff$ f preserves every logical relation

type-indexed family of relations
compatible with type- & term-formers

Question: which relations guarantee definability?

[Plotkin, Jung & Tiuryn, Alimohamed, ...]

f is definable $\iff$ f preserves every logical relation

type-indexed family of relations
compatible with type- & term-formers

Strategy

instantiate general construction with a set $\mathbb{I}$ and an interpretation s.t.

Question: which relations guarantee definability?

[Plotkin, Jung & Tiuryn, Alimohamed, ...]

f is definable $\iff$ f preserves every logical relation

type-indexed family of relations
compatible with type- & term-formers

Strategy

instantiate general construction with a set $\mathbb{I}$ and an interpretation s.t.

1. objects of $\mathrm{OHR}(\mathcal{M})$ are pairs $\left(W, \{R_i \mid i \in \mathbb{I}\}\right)$ + concreteness

Question: which relations guarantee definability?

[Plotkin, Jung & Tiuryn, Alimohamed, ...]

f is definable $\iff$ f preserves every logical relation

type-indexed family of relations
compatible with type- & term-formers

Strategy

instantiate general construction with a set $\mathbb{I}$ and an interpretation s.t.

1. objects of $\mathrm{OHR}(\mathcal{M})$ are pairs $\left(W, \{R_i \mid i \in \mathbb{I}\}\right)$ + concreteness
2. for any logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ there exists $i_0 \in \mathbb{I}$ s.t.

$$\left(\begin{array}{c}\text{relation at index } i_0 \\ \text{for interpretation of } \sigma\end{array}\right) = L_\sigma$$

for every type σ

Strategy

instantiate general construction with a set $\mathbb{I}$ and an interpretation s.t.

1. objects of $\mathrm{OHR}(\mathcal{M})$ are pairs $\left(W, \{R_i \mid i \in \mathbb{I}\}\right)$ + concreteness
2. for any logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ there exists $i_0 \in \mathbb{I}$ s.t.

$$\left(\begin{array}{c} \text{relation at index } i_0 \\ \text{for interpretation of } \sigma \end{array} \right) = L_\sigma$$

Then:

$$\left(\begin{array}{c} f : [\![\Gamma]\!] \to \mathrm{H}\,\hat{T} j [\![\sigma]\!] \\ \text{in } \mathrm{OHR}(\mathcal{M}) \end{array} \right)$$

Strategy

instantiate general construction with a set $\mathbb{I}$ and an interpretation s.t.

1. objects of $\mathrm{OHR}(\mathcal{M})$ are pairs $\left(W, \{R_i \mid i \in \mathbb{I}\}\right)$ + concreteness
2. for any logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ there exists $i_0 \in \mathbb{I}$ s.t.

$$\left(\begin{array}{c}\text{relation at index } i_0 \\ \text{for interpretation of } \sigma\end{array}\right) = L_\sigma$$

Then:

$$\left(\begin{array}{c} f : [\![\Gamma]\!] \to \mathrm{H}\hat{T}j[\![\sigma]\!] \\ \text{in } \mathrm{OHR}(\mathcal{M})\end{array}\right) \iff \left(\begin{array}{c} f \text{ preserves} \\ \text{every relation } R_i\end{array}\right)$$

Strategy

instantiate general construction with a set $\mathbb{I}$ and an interpretation s.t.

1. objects of $\mathrm{OHR}(\mathcal{M})$ are pairs $\left(W, \{R_i \mid i \in \mathbb{I}\}\right)$ + concreteness
2. for any logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ there exists $i_0 \in \mathbb{I}$ s.t.

$$\left(\begin{array}{c} \text{relation at index } i_0 \\ \text{for interpretation of } \sigma \end{array} \right) = L_\sigma$$

Then:

$$\left(\begin{array}{c} f : [\![\Gamma]\!] \to \mathrm{H}\hat{T}j[\![\sigma]\!] \\ \text{in } \mathrm{OHR}(\mathcal{M}) \end{array} \right) \iff \left(\begin{array}{c} f \text{ preserves} \\ \text{every relation } R_i \end{array} \right)$$

$$\implies \left(\begin{array}{c} f \text{ preserves} \\ \text{the logical relation } L \end{array} \right)$$

Strategy

instantiate general construction with a set $\mathbb{I}$ and an interpretation s.t.

1. objects of $\mathrm{OHR}(\mathcal{M})$ are pairs $\left(W, \{R_i \mid i \in \mathbb{I}\}\right)$ + concreteness
2. for any logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ there exists $i_0 \in \mathbb{I}$ s.t.

$$\left(\begin{array}{c}\text{relation at index } i_0 \\ \text{for interpretation of } \sigma\end{array}\right) = L_\sigma$$

Then:

$$\left(\begin{array}{c} f : [\![\Gamma]\!] \to \mathrm{H}\hat{T}j[\![\sigma]\!] \\ \text{in } \mathrm{OHR}(\mathcal{M})\end{array}\right) \iff \left(\begin{array}{c} f \text{ preserves} \\ \text{every relation } R_i\end{array}\right)$$

$$\implies \left(\begin{array}{c} f \text{ preserves} \\ \text{the logical relation } L\end{array}\right)$$

Hence: every map in $\mathrm{OHR}(\mathcal{M})$ is definable

Strategy

instantiate general construction with a set $\mathbb{I}$ and an interpretation s.t.

1. objects of $\mathrm{OHR}(\mathcal{M})$ are pairs $\left(W, \{R_i \mid i \in \mathbb{I}\}\right)$ + concreteness
2. for any logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ there exists $i_0 \in \mathbb{I}$ s.t.

$$\left(\begin{array}{c} \text{relation at index } i_0 \\ \text{for interpretation of } \sigma \end{array} \right) = L_\sigma$$

Then:

$$\left(\begin{array}{c} f : [\![\Gamma]\!] \to \mathrm{H}\hat{T}j[\![\sigma]\!] \\ \text{in } \mathrm{OHR}(\mathcal{M}) \end{array} \right) \iff \left(\begin{array}{c} f \text{ preserves} \\ \text{every relation } R_i \end{array} \right)$$

$$\implies \left(\begin{array}{c} f \text{ preserves} \\ \text{the logical relation } L \end{array} \right)$$

Hence: every map in $\mathrm{OHR}(\mathcal{M})$ is definable

Hence: $\mathrm{OHR}(\mathcal{M})$ is fully abstract

Strategy: suffices for full abstraction!

instantiate general construction with a set $\mathbb{I}$ and an interpretation s.t.

1. objects of $\mathrm{OHR}(\mathcal{M})$ are pairs $\left(W, \{R_i \mid i \in \mathbb{I}\}\right)$
2. for any logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ there exists $i_0 \in \mathbb{I}$ s.t.

$$\left(\begin{array}{c} \text{relation at index } i_0 \\ \text{for interpretation of } \sigma \end{array} \right) = L_\sigma$$

Strategy: suffices for full abstraction!

instantiate general construction with a set $\mathbb{I}$ and an interpretation s.t.

1. objects of $\mathrm{OHR}(\mathcal{M})$ are pairs $\left(W, \{R_i \mid i \in \mathbb{I}\}\right)$ + concreteness
2. for any logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ there exists $i_0 \in \mathbb{I}$ s.t.

$$\left(\begin{array}{c} \text{relation at index } i_0 \\ \text{for interpretation of } \sigma \end{array} \right) = L_\sigma$$

How do we choose $\mathbb{I}$ and $[\![-]\!]$?

Strategy: suffices for full abstraction!

instantiate general construction with a set $\mathbb{I}$ and an interpretation s.t.

1. objects of $\mathrm{OHR}(\mathcal{M})$ are pairs $\left(W, \{R_i \mid i \in \mathbb{I}\}\right)$ + concreteness
2. for any logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ there exists $i_0 \in \mathbb{I}$ s.t.

$$\left(\begin{array}{c} \text{relation at index } i_0 \\ \text{for interpretation of } \sigma \end{array} \right) = L_\sigma$$

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

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$

Strategy: suffices for full abstraction!

instantiate general construction with a set $\mathbb{I}$ and an interpretation s.t.

1. objects of $\mathrm{OHR}(\mathcal{M})$ are pairs $\left(W, \{R_i \mid i \in \mathbb{I}\}\right)$ + concreteness
2. for any logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ there exists $i_0 \in \mathbb{I}$ s.t.

$$\left(\begin{array}{c}\text{relation at index } i_0 \\ \text{for interpretation of } \sigma\end{array}\right) = L_\sigma$$

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

- $\mathbb{I} = \left\{\begin{array}{c}\text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M})\end{array}\right\}$
- Interpretation: i_0 just looks up the required relation

How do we choose $\mathbb{I}$? The intuition:

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$
- Interpretation: i_0 just looks up the required relation

A problem: circular dependencies!

How do we choose $\mathbb{I}$? The intuition:

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$
- Interpretation: i_0 just looks up the required relation

A problem: circular dependencies!

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

How do we choose $\mathbb{I}$? The intuition:

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$
- Interpretation: i_0 just looks up the required relation

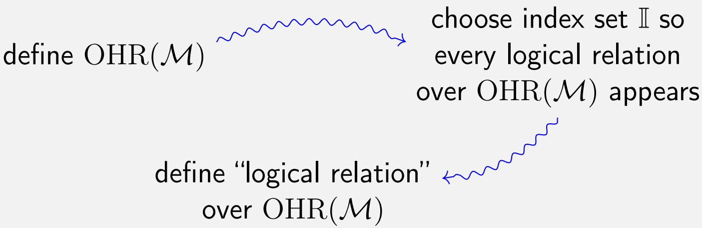
A problem: circular dependencies!

define $\mathrm{OHR}(\mathcal{M})$ $\rightsquigarrow$ choose index set $\mathbb{I}$ so every logical relation over $\mathrm{OHR}(\mathcal{M})$ appears

How do we choose $\mathbb{I}$? The intuition:

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$
- Interpretation: i_0 just looks up the required relation

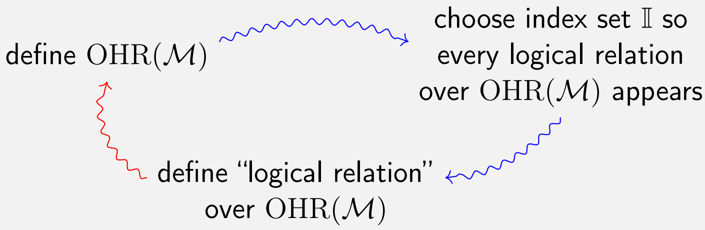
A problem: circular dependencies!



How do we choose $\mathbb{I}$? The intuition:

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$
- Interpretation: i_0 just looks up the required relation

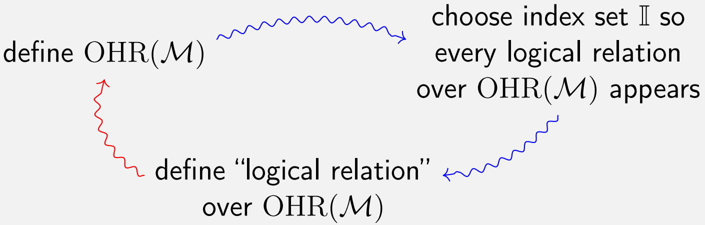
A problem: circular dependencies!



How do we choose $\mathbb{I}$? The intuition:

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$
- Interpretation: i_0 just looks up the required relation

A problem: circular dependencies!
Solution: relations over $\mathrm{OHR}(\mathcal{M})$ are relations over $\mathcal{M}$



How do we choose $\mathbb{I}$? The intuition:

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$
- Interpretation: i_0 just looks up the required relation

A problem: circular dependencies!

Solution: relations over $\mathrm{OHR}(\mathcal{M})$ are relations over $\mathcal{M}$

---

define $\mathrm{OHR}(\mathcal{M})$ ⇝ choose family $\mathbb{I}$ so every possible relation over $\mathcal{M}$ appears ⇝ identify logical relations over $\mathrm{OHR}(\mathcal{M})$ amongst these relations

The OHR construction $\mathrm{OHR}(\mathcal{M})$

objects: $\big(W \in \mathcal{M}, \{R_i \mid i \in \mathbb{I}\}\big)$ + concreteness | maps: maps in $\mathcal{M}$ preserving all R_i

The OHR construction $\mathrm{OHR}(\mathcal{M})$

objects: $\left(W \in \mathcal{M}, \{R_i \mid i \in \mathbb{I}\}\right)$ + concreteness | **maps:** maps in $\mathcal{M}$ preserving all R_i

1. Choose $\mathbb{I}$ 'containing' every relation over $\mathcal{M}$,
   hence every relation over $\mathrm{OHR}(\mathcal{M})$,...

The OHR construction $\mathrm{OHR}(\mathcal{M})$

objects: $(W \in \mathcal{M}, \{R_i \mid i \in \mathbb{I}\})$ + concreteness | maps: maps in $\mathcal{M}$ preserving all R_i

1. Choose $\mathbb{I}$ 'containing' every relation over $\mathcal{M}$,
   hence every relation over $\mathrm{OHR}(\mathcal{M})$,...
2. ...so we can define an interpretation satisfying

$$\exists i_0 \in \mathbb{I}\,.\left(\begin{array}{c}\text{relation at index } i_0 \\ \text{for interpretation of } \beta\end{array}\right) = L_\beta$$

for every logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ and base type β

The OHR construction $\mathrm{OHR}(\mathcal{M})$

objects: $\big(W \in \mathcal{M}, \{R_i \,|\, i \in \mathbb{I}\}\big)$ + concreteness | maps: maps in $\mathcal{M}$ preserving all R_i

1. Choose $\mathbb{I}$ 'containing' every relation over $\mathcal{M}$,
   hence every relation over $\mathrm{OHR}(\mathcal{M})$,...
2. ...so we can define an interpretation satisfying

$$\exists i_0 \in \mathbb{I}\,.\ \left(\begin{array}{c}\text{relation at index } i_0 \\ \text{for interpretation of } \beta\end{array}\right) = L_\beta$$

   for every logical relation $(L_\sigma \,|\, \sigma \in \mathrm{Type})$ and base type β

3. Prove by induction that

$$\left(\begin{array}{c}\text{relation at index } i_0 \\ \text{for interpretation of } \sigma\end{array}\right) = L_\sigma$$

   for all types σ

The OHR construction $\mathrm{OHR}(\mathcal{M})$

objects: $(W \in \mathcal{M}, \{R_i \mid i \in \mathbb{I}\})$ + concreteness | **maps:** maps in $\mathcal{M}$ preserving all R_i

1. Choose $\mathbb{I}$ 'containing' every relation over $\mathcal{M}$,
   hence every relation over $\mathrm{OHR}(\mathcal{M})$,...
2. ...so we can define an interpretation satisfying

$$\exists i_0 \in \mathbb{I}\,.\, \left(\begin{array}{c} \text{relation at index } i_0 \\ \text{for interpretation of } \beta \end{array} \right) = L_\beta$$

   for every logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ and base type β
3. Prove by induction that

$$\left(\begin{array}{c} \text{relation at index } i_0 \\ \text{for interpretation of } \sigma \end{array} \right) = L_\sigma$$

   for all types σ
4. ...hence every map preserves every logical relation,

The OHR construction $\mathrm{OHR}(\mathcal{M})$

objects: $(W \in \mathcal{M}, \{R_i \mid i \in \mathbb{I}\})$ + concreteness | maps: maps in $\mathcal{M}$ preserving all R_i

1. Choose $\mathbb{I}$ 'containing' every relation over $\mathcal{M}$,
   hence every relation over $\mathrm{OHR}(\mathcal{M})$,...
2. ...so we can define an interpretation satisfying

   $$\exists i_0 \in \mathbb{I} \,.\, \left(\begin{array}{c} \text{relation at index } i_0 \\ \text{for interpretation of } \beta \end{array} \right) = L_\beta$$

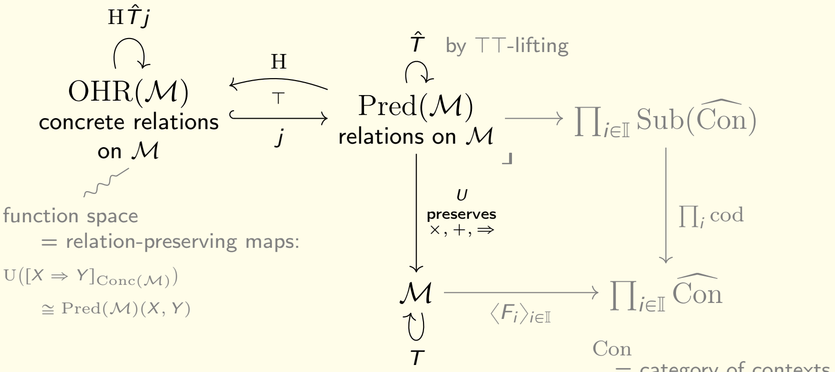
   for every logical relation $(L_\sigma \mid \sigma \in \mathrm{Type})$ and base type β
3. Prove by induction that

   $$\left(\begin{array}{c} \text{relation at index } i_0 \\ \text{for interpretation of } \sigma \end{array} \right) = L_\sigma$$

   for all types σ
4. ...hence every map preserves every logical relation,
5. ...hence every map is definable & the model is fully abstract

The OHR construction

Choose $\mathbb{I}$ as above, then instantiate general construction as follows:



Codomain fibration on presheaves

$\rightsquigarrow$ relations are Kripke relations of varying arity

This work

This work

1. A construction that takes a signature and a well-pointed model and returns a fully abstract model

This work

1. A construction that takes a signature and a well-pointed model and returns a fully abstract model
2. Holds over Set whenever there's a name for every $x \in [\![\beta]\!]$.

This work

1. A construction that takes a signature and a well-pointed model and returns a fully abstract model
2. Holds over Set whenever there's a name for every $x \in [\![\beta]\!]$.

Key ideas

This work

1. A construction that takes a signature and a well-pointed model and returns a fully abstract model
2. Holds over Set whenever there's a name for every $x \in [\![\beta]\!]$.

Key ideas

1. Cut out bad maps using the general construction: pair objects with families of concrete relations

This work

1. A construction that takes a signature and a well-pointed model and returns a fully abstract model
2. Holds over Set whenever there's a name for every $x \in [\![\beta]\!]$.

Key ideas

1. Cut out bad maps using the general construction: pair objects with families of concrete relations
2. Preserving every logical relation $\implies$ ensures definability

This work

1. A construction that takes a signature and a well-pointed model and returns a fully abstract model
2. Holds over Set whenever there's a name for every $x \in [\![\beta]\!]$.

Key ideas

1. Cut out bad maps using the general construction: pair objects with families of concrete relations
2. Preserving every logical relation $\implies$ ensures definability
3. Avoid circularity by choosing indexing set $\mathbb{I}$ carefully.

This work

1. A construction that takes a signature and a well-pointed model and returns a fully abstract model
2. Holds over Set whenever there's a name for every $x \in [\![\beta]\!]$.

Key ideas

1. Cut out bad maps using the general construction: pair objects with families of concrete relations
2. Preserving every logical relation $\implies$ ensures definability
3. Avoid circularity by choosing indexing set $\mathbb{I}$ carefully.

Still to do

1. Weaken assumptions: well-pointedness, hull functor H, ...
2. Enrichment $\rightsquigarrow$ recover recursion?
3. Universal property?

This work

1. A construction that takes a signature and a well-pointed model and returns a fully abstract model
2. Holds over Set whenever there's a name for every $x \in [\![\beta]\!]$.

Key ideas

1. Cut out bad maps using the general construction: pair objects with families of concrete relations
2. Preserving every logical relation $\implies$ ensures definability
3. Avoid circularity by choosing indexing set $\mathbb{I}$ carefully.

Still to do

`philip.saville@cs.ox.ac.uk`

1. Weaken assumptions: well-pointedness, hull functor H, ...
2. Enrichment $\rightsquigarrow$ recover recursion?
3. Universal property?

How do we choose $\mathbb{I}$? The circular intuition:

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$
- Interpretation: i_0 just looks up the required relation

How do we choose $\mathbb{I}$? The circular intuition:

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$
- Interpretation: i_0 just looks up the required relation

Avoid circularity! Quantify over enough relations so that

- $i \in \mathbb{I}$ is a tuple $(\ldots, R, \ldots)$ with R a relation

How do we choose $\mathbb{I}$? The circular intuition:

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$
- Interpretation: i_0 just looks up the required relation

Avoid circularity! Quantify over enough relations so that

- $i \in \mathbb{I}$ is a tuple $(\ldots, R, \ldots)$ with R a relation
- choose the semantic interpretation

$$\left(\begin{array}{c} \text{carrier of } [\![\beta]\!] \\ \text{in } \mathrm{OHR}(\mathcal{M}) \end{array} \right) := \left(\begin{array}{c} \text{interpretation} \\ \text{of } \beta \text{ in } \mathcal{M} \end{array} \right)$$

$$\left(\begin{array}{c} \text{relation at index} \\ (\ldots, R, \ldots) \text{ for } [\![\beta]\!] \end{array} \right) := R$$

How do we choose $\mathbb{I}$? The circular intuition:

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$
- Interpretation: i_0 just looks up the required relation

Avoid circularity! Quantify over enough relations so that

- $i \in \mathbb{I}$ is a tuple $(\ldots, R, \ldots)$ with R a relation
- choose the semantic interpretation

$$\left(\begin{array}{c} \text{carrier of } [\![\beta]\!] \\ \text{in } \mathrm{OHR}(\mathcal{M}) \end{array} \right) := \left(\begin{array}{c} \text{interpretation} \\ \text{of } \beta \text{ in } \mathcal{M} \end{array} \right)$$

$$\left(\begin{array}{c} \text{relation at index} \\ (\ldots, R, \ldots) \text{ for } [\![\beta]\!] \end{array} \right) := R$$

- choose $i_0 := (\ldots, L_\beta, \ldots)$;

How do we choose $\mathbb{I}$? The circular intuition:

- $\mathbb{I} = \left\{ \begin{array}{c} \text{set of logical relations} \\ \text{over } \mathrm{OHR}(\mathcal{M}) \end{array} \right\}$
- Interpretation: i_0 just looks up the required relation

Avoid circularity! Quantify over enough relations so that

- $i \in \mathbb{I}$ is a tuple $(\ldots, R, \ldots)$ with R a relation
- choose the semantic interpretation

$$\left(\begin{array}{c} \text{carrier of } [\![\beta]\!] \\ \text{in } \mathrm{OHR}(\mathcal{M}) \end{array} \right) := \left(\begin{array}{c} \text{interpretation} \\ \text{of } \beta \text{ in } \mathcal{M} \end{array} \right)$$

$$\left(\begin{array}{c} \text{relation at index} \\ (\ldots, R, \ldots) \text{ for } [\![\beta]\!] \end{array} \right) := R$$

- choose $i_0 := (\ldots, L_\beta, \ldots)$; prove for all σ by induction