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

Citation for published version:
Kammar, O, Katsumata, SY & Saville, P 2022, 'Fully abstract models for effectful -calculi via category-theoretic logical relations', *Proceedings of the ACM on Programming Languages*, vol. 6, no. POPL, 44. https://doi.org/10.1145/3498705

Digital Object Identifier (DOI):
10.1145/3498705

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Proceedings of the ACM on Programming Languages

Fully Abstract Models for Effectful λ-Calculi via Category-Theoretic Logical Relations[*]

OHAD KAMMAR, University of Edinburgh, U.K.
SHIN-YA KATSUMATA, National Institute of Informatics, Japan
PHILIP SAVILLE, University of Oxford, U.K.

We present a construction which, under suitable assumptions, takes a model of Moggi's computational λ-calculus with sum types, effect operations and primitives, and yields a model that is adequate and fully abstract. The construction, which uses the theory of fibrations, categorical glueing, $\top\top$-lifting, and $\top\top$-closure, takes inspiration from O'Hearn & Riecke's fully abstract model for PCF. Our construction can be applied in the category of sets and functions, as well as the category of diffeological spaces and smooth maps and the category of quasi-Borel spaces, which have been studied as semantics for differentiable and probabilistic programming.

CCS Concepts: • **Theory of computation** → **Denotational semantics**; **Categorical semantics**.

Additional Key Words and Phrases: full abstraction, call-by-value, O'Hearn & Riecke, fibration, monad

ACM Reference Format:
Ohad Kammar, Shin-ya Katsumata, and Philip Saville. 2022. Fully Abstract Models for Effectful λ-Calculi via Category-Theoretic Logical Relations. *Proc. ACM Program. Lang.* 6, POPL, Article 44 (January 2022), 28 pages. https://doi.org/10.1145/3498705

1 INTRODUCTION

Two programs are *contextually equivalent* if they evaluate to the same result (with the same effect) in all program contexts. This natural notion of equality makes precise the intuitive 'sameness' a programmer is generally interested in, for example for optimization or compilation. However, the quantification over all program contexts makes establishing contextual equivalence notoriously difficult. A wide variety of techniques have been proposed to mitigate this difficulty: in this work we focus on a particular strand, namely the construction of *fully abstract denotational models.*

A *denotational semantics* assigns a mathematical object (*e.g.* a set-theoretic function) to each term or program construct. Two terms are *denotationally equal* if they are assigned the same denotation. If any two terms with the same denotation are contextually equivalent, a model is *adequate*; if any two contextually-equivalent terms have the same denotation, it is *fully abstract.* Thus, adequacy and full abstraction roughly correspond to soundness and completeness in logic: in an adequate and fully abstract model, denotational equality completely characterises contextual equivalence. By constructing fully abstract models, one reduces contextual equivalence to denotational equality.

[*] PS was supported by the Air Force Office of Scientific Research under award number FA9550-21-1-0038. OK and PS were supported by Royal Society University Research Fellow Enhancement Award number RGF/EA/181059. SK was supported by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603).

Authors' addresses: Ohad Kammar, School of Informatics, University of Edinburgh, U.K., ohad.kammar@ed.ac.uk; Shin-ya Katsumata, National Institute of Informatics, Japan, s-katsumata@nii.ac.jp; Philip Saville, Department of Computer Science, University of Oxford, U.K., philip.saville@cs.ox.ac.uk.

2475-1421/2022/1-ART44
https://doi.org/10.1145/3498705

In this paper we construct fully abstract models for a wide class of languages with *effects* encoded by monads (à la Moggi [1989, 1991]), including set-theoretic state (Sec. 9), measure-theoretic probabilistic programming (Sec. 11.1) and state in differentiable neural network programming (Sec. 11.2). Our main theorem (Thm. 10.2) says the following. Suppose one chooses an effect (*e.g.* global state) and a *signature* of operations (*e.g.* operations for reading and writing to memory), as well as a *semantic model* consisting of a category $\mathcal{M}$, a monad, and a denotation for each operation. Then, subject to suitable conditions on $\mathcal{M}$, one can construct an adequate and fully abstract model in which the morphisms are maps in $\mathcal{M}$ preserving certain predicates. Our construction is inspired by that of O'Hearn and Riecke [1995], so we call it the *OHR construction.*

1.1 First Steps towards the OHR Construction

Much of the preceding work on full abstraction has focussed on languages with recursion (see Sec. 1.3), but even set-theoretic models for simple languages can fail to be fully abstract. The next example, which we learned from C. Matache & S. Staton, shows the natural model of read-only state for a global one-bit reference cell is not fully abstract. For now we use notation rather loosely.

Example 1.1. Consider a language with a type bool of booleans and a signature containing terms tt and ff for the booleans, an operation read of type $(1 \to \mathrm{bool})$ reading from the reference cell, and operations for the usual logical operations on booleans. This language has a natural semantic interpretation in the category **Fin** of finite sets and functions. We take **Fin** rather than the large category of **Set** of all sets and functions to avoid size issues later on (see Sec. 8); this does not affect the model. On bool, set $[\![\mathrm{bool}]\!] := \mathbb{2} = \{0, 1\}$. Programs are interpreted using the *reader monad* $\mathrm{R}X := (\mathbb{2} \Rightarrow X)$: a term $(\Gamma \vdash M : \sigma)$ is denoted by a function $[\![\Gamma]\!] \to \mathrm{R}[\![\sigma]\!]$ whose values are functions from the state of the read-only bit to the end result. Product types are interpreted using the cartesian product of sets, and arrow types by sets of functions: $[\![\sigma \to \tau]\!] := ([\![\sigma]\!] \Rightarrow \mathrm{R}[\![\tau]\!])$. In particular, the denotation of a closed program of type σ may be identified with an element of $\mathrm{R}[\![\sigma]\!]$.

The idea is that programs are parametrised by the value of the reference cell. For example, $[\![\mathrm{read}\,()]\!] = \mathrm{id}_{\mathbb{2}}$: if the reference cell contains i, the read operation returns i. Similarly, the program $\big((\mathrm{read}\,()) \text{ or } \neg(\mathrm{read}\,())\big)$ reading from the state twice, negating one result, and taking the disjunction, is interpreted by const_1, the constant function at 1: the result is 1 no matter what is stored in the cell.

This defines a semantic model: the category is **Fin**, the monad is R, and operations are interpreted as sketched above. Now consider the following closed terms of type $((1 \to \mathrm{bool}) \to \mathrm{bool}) \to \mathrm{bool}$:

$$M := \lambda f.(f\, \lambda x.\mathrm{tt}) \text{ or } (f\, \lambda x.\mathrm{ff}) \qquad \text{and} \qquad M' := \lambda f.(f\, \mathrm{read}) \text{ or } \big(f\, (\lambda x.\neg(\mathrm{read}\,()))\big) \tag{1}$$

Semantically, M and M' are interpreted by elements of $\mathrm{R}\big(\big((1 \Rightarrow \mathrm{R}\mathbb{2}) \Rightarrow \mathrm{R}\mathbb{2}\big) \Rightarrow \mathrm{R}\mathbb{2}\big)$. As the state can only be read, and never changed, it is intuitively clear that M and M' are contextually equivalent (we give a proof using our construction in Lemma 9.1). However, $[\![M]\!] \neq [\![M']\!]$: if we define $\kappa : (1 \Rightarrow \mathrm{R}\mathbb{2}) \to \mathrm{R}\mathbb{2}$ by $\kappa(g) = \mathrm{const}_1$ if $g(*) = \mathrm{const}_1$ and $\kappa(g) = \mathrm{const}_0$ otherwise, then for any $i, j \in \mathbb{2}$ one has $[\![M]\!](i)(\kappa)(j) = 1$ but $[\![M']\!](i)(\kappa)(j) = 0$.

This example highlights the main obstacle to constructing fully abstract models. Typically, a denotational model contains "counterexamples to contextual equivalence": morphisms such as κ which can be used to distinguish the denotations of contextually equivalent terms. A classic example is the *parallel-or* function, which is commonly used to show the domains model of PCF is not fully abstract [Plotkin 1977]. To obtain a fully abstract model from a non-fully-abstract one, therefore, we need to remove all the counterexamples to contextual equivalence.

How might we refine Ex. 1.1 to remove κ? Intuitively, κ cannot correspond to a program because it uses too much information. To compute $\kappa(g)$ one must verify that $g(*)$ returns 1 both when the reference cell contains 0 and when the reference cell contains 1. In other words, κ must know how g

behaves in every possible state. But the state is read-only, so programs cannot do this. This suggests that we should restrict to morphisms which do not use such extra information. One way to do this is to construct a new, refined model in which objects are paired with relations and maps are required to preserve these relations. In Ex. 1.2 we outline such a model, again due to C. Matache & S. Staton. Instead of taking just sets, the objects are sets paired with relations R_0 and R_1 which constrain the behaviour of morphisms when the reference cell contains 0 and 1, respectively.

Example 1.2. Let $\mathbb{L}_{\mathbf{Fin}}$ be the category with objects given by triples $(\underline{X}, R_0, R_1)$, where $\underline{X} \in \mathbf{Fin}$ and $R_0, R_1 \subseteq \underline{X}^2$ are binary relations on $\underline{X}$, and morphisms $f : (\underline{X}, R_0, R_1) \to (\underline{Y}, S_0, S_1)$ given by maps $f : \underline{X} \to \underline{Y}$ on the carrier sets which preserve both relations: if $(x, x') \in R_i$ then $(fx, fx') \in S_i$. This category has enough structure to model the calculus: it is cartesian closed and has a (strong) monad $\hat{\mathrm{R}}$ defined by $\hat{\mathrm{R}}(\underline{X}, R_0, R_1) := \big(\mathrm{R}\underline{X}, \hat{\mathrm{R}}(R_0), \hat{\mathrm{R}}(R_1)\big)$, where $(h, h') \in \hat{\mathrm{R}}(R_i)$ if and only if $(h\,i, h'\,i) \in R_i$.

We give a denotational semantics to programs in this category by interpreting bool as $l(\mathrm{bool}) := \big(\mathbb{2}, \{(0,0),(1,1)\}, \{(0,0),(1,1)\}\big)$; this extends to a semantic interpretation of every type using the cartesian closed structure, and the interpretations of the primitives and operations in **Fin** all define morphisms in $\mathbb{L}_{\mathbf{Fin}}$ with respect to this new structure. Then one obtains an interpretation in $\mathbb{L}_{\mathbf{Fin}}$ of every term using the cartesian closed structure and the monad $\hat{\mathrm{R}}$. We denote the interpretation of $(\Gamma \vdash M : \sigma)$ in $\mathbb{L}_{\mathbf{Fin}}$ by $l\llbracket \Gamma \vdash M : \sigma \rrbracket$ to distinguish it from the interpretation in **Fin**.

Now, κ is not a morphism $l\llbracket 1 \to \mathrm{bool} \rrbracket \to \hat{\mathrm{R}}(l\llbracket \mathrm{bool} \rrbracket)$ in $\mathbb{L}_{\mathbf{Fin}}$. Writing R_0^σ and R_1^σ for the two relations in $l\llbracket \sigma \rrbracket$, then $\big(\lambda x\,.\,\mathrm{id}_{\mathbb{2}}, \lambda x\,.\,\mathrm{const}_1\big) \in R_1^{1\to\mathrm{bool}}$ but $\big(\kappa(\lambda x\,.\,\mathrm{id}_{\mathbb{2}}), \kappa(\lambda x\,.\,\mathrm{const}_1)\big) = (\mathrm{const}_0, \mathrm{const}_1)$ is not in R_1^{bool}. In fact, this is sufficient to show κ is not the denotation of any term in the **Fin**-model of Ex. 1.1. To see this, note the forgetful functor $\mathrm{U} : \mathbb{L}_{\mathbf{Fin}} \to \mathbf{Fin} : (\underline{X}, R_0, R_1) \mapsto \underline{X}$ strictly preserves the semantic interpretation: $\mathrm{U}(l\llbracket \Gamma \vdash M : \sigma \rrbracket) = \llbracket \Gamma \vdash M : \sigma \rrbracket$ for every term. If κ were definable, so that $\kappa = \llbracket x : 1 \to \mathrm{bool} \vdash K : \mathrm{bool} \rrbracket$, we would get $\kappa = \llbracket x : 1 \to \mathrm{bool} \vdash K : \mathrm{bool} \rrbracket = \mathrm{U}(l\llbracket x : 1 \to \mathrm{bool} \vdash K : \mathrm{bool} \rrbracket)$. Since $\mathrm{U}(f) = f$, this entails κ is a map in $\mathbb{L}_{\mathbf{Fin}}$, a contradiction.

The model just sketched is an improvement on that in Ex. 1.1: we have removed the counterexample to contextual equivalence κ. However, this success is only partial. Even though κ is not a morphism in $\mathbb{L}_{\mathbf{Fin}}$, it is still an element of the function space. Since $\kappa \in \llbracket (1 \to \mathrm{bool}) \to \mathrm{bool} \rrbracket$ in **Fin** and the forgetful functor preserves the semantic interpretation, $\kappa \in \mathrm{U}(l\llbracket (1 \to \mathrm{bool}) \to \mathrm{bool} \rrbracket)$. But then $\mathrm{U}(l\llbracket M \rrbracket)(i)(\kappa) = \llbracket M \rrbracket(i)(\kappa) \neq \llbracket M' \rrbracket(i)(\kappa) = \mathrm{U}(l\llbracket M' \rrbracket)(i)(\kappa)$. Since U is faithful, it follows that $l\llbracket M \rrbracket \neq l\llbracket M' \rrbracket$: even though we've cut κ out of our semantic model, its existence in the function space is sufficient to distinguish the denotations of contextually equivalent terms. We solve this using the notion of *concreteness* (*cf.* O'Hearn and Riecke [1995]).

1.1.1 Cutting Down the Function Space: Concreteness. We start by expressing the property "κ is not a morphism in $\mathbb{L}_{\mathbf{Fin}}$ but it is an element of the function space" in categorical terms.

Lemma 1.1. *The set map* $\ulcorner \kappa \urcorner : 1 \to \llbracket (1 \to \mathrm{bool}) \to \mathrm{bool} \rrbracket : * \mapsto \kappa$ *does not define a morphism from the terminal object to* $l\llbracket (1 \to \mathrm{bool}) \to \mathrm{bool} \rrbracket$ *in* $\mathbb{L}_{\mathbf{Fin}}$.

This lemma suggests that we need to restrict our function spaces to consist only of those elements that are 'named' by a a morphism from the terminal object (a *global element*) in $\mathbb{L}_{\mathbf{Fin}}$. We do this by restricting our attention to the subcategory of *concrete* objects.

Definition 1.1. An object $X := (\underline{X}, R_0, R_1) \in \mathbb{L}_{\mathbf{Fin}}$ is *concrete* if for every $x \in \underline{X}$ the corresponding global element $\ulcorner x \urcorner : 1 \to \underline{X}$ in **Fin** lifts to a global element $\ulcorner x \urcorner : 1 \to X$ in $\mathbb{L}_{\mathbf{Fin}}$.

Explicitly, $(\underline{X}, R_0, R_1)$ is concrete if for every $x \in \underline{X}$ the pair (x, x) is in both R_0 and R_1. Write $\mathbb{C}_{\mathbf{Fin}}$ for the full subcategory of $\mathbb{L}_{\mathbf{Fin}}$ consisting of just the concrete objects. This category has enough structure to be a semantic model because it is a *reflective and coreflective subcategory* of $\mathbb{L}_{\mathbf{Fin}}$: the inclusion $j : \mathbb{C}_{\mathbf{Fin}} \hookrightarrow \mathbb{L}_{\mathbf{Fin}}$ has both left and right adjoints. The left adjoint K adds the diagonal:

$$\mathrm{K}(\underline{X}, R_0, R_1) := \left(\underline{X}, R_0 \cup \{(x,x) \mid x \in \underline{X}\}, R_1 \cup \{(x,x) \mid x \in \underline{X}\}\right) \tag{2}$$

The right adjoint H, by contrast, cuts down the carrier set. It takes $(\underline{X}, R_0, R_1)$ to the object with carrier $\{x \in \underline{X} \mid (x,x) \in R_0 \text{ and } (x,x) \in R_1\}$ and relations given by restricting R_0 and R_1 to this set.

We can now give the structure for the semantic model. The interpretation of bool in $\mathbb{L}_{\mathbf{Fin}}$ is concrete, so the interpretation of base types and constants in $\mathbb{L}_{\mathbf{Fin}}$ restricts to $\mathbb{C}_{\mathbf{Fin}}$. Next, the monad $\hat{\mathrm{R}}$ induces a strong monad on $\mathbb{C}_{\mathbf{Fin}}$ with underlying functor $\mathrm{H}\hat{\mathrm{R}}j$ via the adjunction $j \dashv \mathrm{H}$. Finally, by the general theory of (co)reflective subcategories, $\mathbb{C}_{\mathbf{Fin}}$ is a cartesian closed category. Products are computed as in $\mathbb{L}_{\mathbf{Fin}}$ but exponentials are not: the function space $(X \Rightarrow_{\mathbb{C}_{\mathbf{Fin}}} Y)$ in $\mathbb{C}_{\mathbf{Fin}}$ is obtained by applying H to the function space $(jX \Rightarrow_{\mathbb{L}_{\mathbf{Fin}}} jY)$ in $\mathbb{L}_{\mathbf{Fin}}$. Thus, the function space in $\mathbb{C}_{\mathbf{Fin}}$ is a version of that in $\mathbb{L}_{\mathbf{Fin}}$ which has been 'cut down' by H. The next result makes this explicit.

Lemma 1.2. *For any $X, Y \in \mathbb{C}_{\mathbf{Fin}}$ the carrier set of $(X \Rightarrow_{\mathbb{C}_{\mathbf{Fin}}} Y)$ is isomorphic to $\mathbb{L}_{\mathbf{Fin}}(jX, jY)$.*

The lemma says that we may identify elements of the function space $(X \Rightarrow_{\mathbb{C}_{\mathbf{Fin}}} Y)$ with morphisms in $\mathbb{L}_{\mathbf{Fin}}$. In particular, unwinding the isomorphism shows that, since κ is not a morphism in $\mathbb{L}_{\mathbf{Fin}}$, it cannot be an element of the function space in $\mathbb{C}_{\mathbf{Fin}}$. This also explains why the forgetful functor $\mathrm{U} : \mathbb{C}_{\mathbf{Fin}} \to \mathbf{Fin}$ cannot preserve exponentials—and hence the semantic interpretation—even though it does preserve products: in general $\mathbb{L}_{\mathbf{Fin}}(X, Y) \subsetneq \mathbf{Fin}(\underline{X}, \underline{Y}) = (\underline{X} \Rightarrow \underline{Y})$.

Remark 1.1. The requirement that U does not preserve the semantic interpretation is necessary. If $\mathbb{D}$ is a fully abstract semantic model and the functor $F : \mathbb{D} \to \mathbf{Fin}$ preserves the semantic interpretation, then the denotations in $\mathbb{D}$ of the terms M and M' in (1) must be equal (by full abstraction) but their images under F, namely $[\![M]\!]$ and $[\![M']\!]$, are not (by Ex. 1.2): a contradiction.

One might hope that the semantic model on $\mathbb{C}_{\mathbf{Fin}}$ is fully abstract. However, this is not the case. Although we have cut κ out of both the hom-sets and the function spaces, other counterexamples to contextual equivalence remain: the notion of 'relation' employed in $\mathbb{L}_{\mathbf{Fin}}$ is too weak. Nonetheless, the construction of $\mathbb{C}_{\mathbf{Fin}}$ highlights the two sufficient conditions we will rely on to obtain full abstraction in our OHR construction, and provides a template for how to go about ensuring them.

1.1.2 Sufficient Conditions for Full Abstraction. Our construction of $\mathbb{C}_{\mathbf{Fin}}$ had two stages: first, to remove the counterexample to contextual equivalence κ from being a morphism, then to remove it from the function space. Each stage corresponds to a condition we shall require for full abstraction.

First we want to prevent any counterexamples to contextual equivalence from being morphisms. Intuitively, such counterexamples are morphisms that provide information which is not available within the syntax. It is therefore plausible that if every map $[\![\Gamma]\!] \to T[\![\sigma]\!]$ is *definable*—that is, the denotation of some term—then no counterexamples to contextual equivalence can exist. A semantic model satisfying this property is called *fully complete* [Abramsky and Jagadeesan 1994].

Next we want to phrase the property achieved by concreteness as a condition on the underlying category. One way to describe the problem caused by Lemma 1.1 is that in $\mathbb{L}_{\mathbf{Fin}}$ we can have $l[\![M]\!] \neq l[\![M']\!]$ even though $l[\![M]\!] \circ g = l[\![M']\!] \circ g$ for every global element $g : 1 \to l[\![(1 \to \mathrm{bool}) \to \mathrm{bool}]\!]$. Hence $l[\![M]\!]$ and $l[\![M']\!]$ may agree on the denotation of every closed term but still not be equal. This is the property we want to outlaw: we require our semantic model to be *well-pointed*, so two maps $f, f' : X \to Y$ are equal iff $f \circ g = f' \circ g$ for every $g : 1 \to X$ (*cf.* Freyd [1972]; Lawvere [2006]). Indeed, the step from $\mathbb{L}_{\mathbf{Fin}}$ to $\mathbb{C}_{\mathbf{Fin}}$ restricts a non-well-pointed category to a well-pointed one.

Even though we have only provided informal motivation, full completeness and well-pointedness are actually sufficient to guarantee full abstraction: see Prop. 3.1.

$$\begin{tikzcd}
\mathbb{C}_{\mathbf{Fin}} \arrow[loop above, "\mathrm{H}\hat{\mathrm{R}}j"] \arrow[r] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] & \mathbb{L}_{\mathbf{Fin}} \arrow[loop above, "\hat{\mathrm{R}}"] \arrow[l, bend right, "\mathrm{K}"'] \arrow[l, bend left, "\mathrm{H}"] \arrow[r] \arrow[d, "\mathrm{U}"'] \arrow[dr, phantom, very near start, "\lrcorner"] & \mathrm{Sub}(\mathbf{Fin}) \times \mathrm{Sub}(\mathbf{Fin}) \arrow[d, "\mathrm{cod} \times \mathrm{cod}"] \\
& \mathbf{Fin} \arrow[loop left, "\mathrm{R}"] \arrow[r, "\Delta"'] & \mathbf{Fin} \times \mathbf{Fin}
\end{tikzcd}$$

(a) The construction of $\mathbb{C}_{\mathbf{Fin}}$

$$\begin{tikzcd}
\mathcal{C} \arrow[loop above, "\mathrm{W}:=\mathrm{H}\hat{T}j"] \arrow[r] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] & \mathcal{K} \arrow[loop above, "\hat{T}"] \arrow[l, bend right, "\mathrm{K}"'] \arrow[l, bend left, "\mathrm{H}"] \arrow[r] \arrow[d, "\mathrm{U}"'] \arrow[dr, phantom, very near start, "\lrcorner"] & \prod_{i \in \mathbb{I}} \mathrm{Sub}(\widehat{\mathbb{C}_i}) \arrow[d, "\prod_{i \in \mathbb{I}} \mathrm{cod}"] \\
& \mathcal{M} \arrow[loop left, "T"] \arrow[r, "\langle \mathrm{N}_{F_i} \rangle_{i \in \mathbb{I}}"'] & \prod_{i \in \mathbb{I}} \widehat{\mathbb{C}_i}
\end{tikzcd}$$

(b) The OHR construction

Fig. 1. The construction of $\mathbb{C}_{\mathbf{Fin}}$ compared to the OHR construction

1.1.3 A Template for the OHR Construction. Even though $\mathbb{C}_{\mathbf{Fin}}$ is not fully abstract, it does highlight the two key steps our OHR construction will take. We therefore finish our exploration of $\mathbb{C}_{\mathbf{Fin}}$ by presenting its construction in abstract terms. At this stage the details of the definitions are not important: instead one should focus on the broad outline, because the OHR construction will have the same shape. For a more detailed, but still high-level, overview see Sec. 2.

$\mathbb{L}_{\mathbf{Fin}}$ may be constructed using *fibrations for logical relations* [Katsumata 2005, 2013]. These build on the observation of Hermida [1993] and Jacobs [1999] that properties of programming languages are naturally described using (Grothendieck) *fibrations*, and axiomatise the structure required to study logical relations in category-theoretic terms. At this stage our main example arises from the *subobject fibration* $\mathrm{cod} : \mathrm{Sub}(\mathbf{Fin}) \to \mathbf{Fin}$. The objects of $\mathrm{Sub}(\mathbf{Fin})$ are pairs of finite sets (X, A) such that $A \subseteq X$: we think of A as a unary predicate on X. Morphisms $(X, A) \to (X', A')$ are functions $f : X \to X'$ which preserve the predicates. The functor cod takes (X, A) to the superset X.

One obtains $\mathbb{L}_{\mathbf{Fin}}$ by *change-of-base* along the functor $\Delta : X \mapsto (X, X) : \mathbf{Fin} \to \mathbf{Fin} \times \mathbf{Fin}$. This means $\mathbb{L}_{\mathbf{Fin}}$ is the category constructed as the pullback in Figure 1a. The theory of fibrations for logical relations guarantees that $\mathbb{L}_{\mathbf{Fin}}$ is cartesian closed and that U strictly preserves this structure. The construction is completed by restricting to the full subcategory of concrete objects, as shown. Although we have yet to give the relevant definitions—we sketch these in Sec. 2—Figure 1b shows our OHR construction takes the same form. The category C is constructed by choosing an appropriate fibration for logical relations, applying change-of-base, then restricting to the full subcategory of concrete objects. Choosing $\mathbb{I}$ appropriately gives the OHR model $\textsc{ohr}(\mathcal{M})$.

1.2 This Paper

We construct a fully abstract model for λ_c^+, an extension of Moggi's computational λ-calculus with sum types, primitives, and effect operations. Our main result is that, for any model of λ_c^+ satisfying suitable conditions, the OHR construction over that model exists and is fully abstract (Thm. 10.2). We take an abstract, category-theoretic approach so that our construction is parametric in the input model and works for any choice of monadic effect.

We start without sum types (Secs. 7 and 8), then refine the construction to include them (Sec. 10). As in O'Hearn & Riecke's work, the maps in $\textsc{ohr}(\mathcal{M})$ are maps in $\mathcal{M}$ preserving a certain class of relations, the carrier of the function space is a sub-space of the function space in $\mathcal{M}$, and the canonical functor $\textsc{ohr}(\mathcal{M}) \to \mathcal{M}$ strictly preserves sums and products (but *not* exponentials, *cf.* Remark 1.1). Thus, $\mathcal{M}$ and $\textsc{ohr}(\mathcal{M})$ are tightly connected: we explore this in Sec. 9. Our development makes use of the abstract framework of *fibrations for logical relations* and *logical relations of varying arity* (Sec. 4), as well as an analysis of the definability predicate for λ_c^+ over an arbitrary model (Sec. 5). We also give a bird's-eye view of the key steps in the construction before diving into the details (Sec. 2).

As applications, we instantiate the construction for three different languages, each over a different category of sets-with-structure (Secs. 9 and 11). We consider the language for read-only state of

Ex. 1.1, an idealised language for probabilistic programming over the category of quasi-Borel spaces [Heunen et al. 2017]), and a simple language for differential programming over the category of diffeological spaces [Huot et al. 2020; Iglesias-Zemmour 2013; Souriau 1980]. In Sec. 9 we also provide a conceptual justification for why the counterexample κ of Ex. 1.1 does not give rise to a counterexample in the OHR model. As well as showcasing the structure of the OHR model, and how one can work with it, this helps explain why the construction succeeds.

Technical Contributions. The key message of this paper is that fully abstract models for λ_c^+ can often be constructed mechanistically, without using special features of the model or language. Indeed, our construction applies to any set-theoretic model of λ_c^+, independently of the monadic effect, so long as every element of the base types is denoted by some term (see Ex. 8.1).

The main technical contributions are as follows. (1) An analysis of the relationship between logical relations, semantic-interpretation preserving functors, and definability. This includes a characterisation of the definable morphisms in an arbitrary model of λ_c^+, which builds on the work of Katsumata [2008, 2013] (Sec. 5). (2) The introduction of an abstract, fibrational notion of *concreteness*, which serves to 'cut down' a function space to just those morphisms preserving suitable properties (Sec. 6). (3) A conceptual, category-theoretic framework for constructing fully abstract models independently of the choice of monadic effect, base types, and primitives (Thm. 10.2). As well as covering a wide range of examples, this also lays the foundations for future work.

1.3 Related Work

Our analysis of definability—and logical relations more generally—builds on Katsumata's work on sum types (2008) and computational effects (2005). The theory we develop shows how to combine these two approaches, and also extends from the monadic metalanguage over **Set** to the computational λ-calculus over an arbitrary model. As far as we are aware, this is the first characterisation of definability for the computational λ-calculus, with or without sum types. A very different approach is due to Fiore and Simpson [1999], who characterise definability for sum types in the simply-typed λ-calculus using a sheaf condition. The closest work to our own is that of Goubault-Larrecq et al. [2004], who show that a certain logical relation is sound and complete for contextual equivalence for a version of Moggi's monadic metalanguage with cryptographic primitives and name generation. This builds on previous work [Lasota et al. 2007], which also relates the monadic lifting of Goubault-Larrecq et al. [2008] to logical relations, and hence a form of full abstraction, but only for specific effects and types up to first-order.

Because morphisms in $\textsc{ohr}(\mathcal{M})$ are maps in $\mathcal{M}$ which preserve certain relations, one can test equality of morphisms in $\textsc{ohr}(\mathcal{M})$ just as easily as in $\mathcal{M}$: for example, over a set-theoretic model it is just equality of functions. This contrasts with the fully abstract models constructed by quotienting with an equivalence relation generated from the syntax (*e.g.* [de' Liguoro 1996; Milner 1977]), where morphisms are equivalence classes of maps in the original model. Indeed, writing $\mathrm{deL}[\![\sigma]\!]$ for the interpretation of a type in a de'Liguoro-style model and $\mathcal{M}[\![\sigma]\!]$ for its interpretation in the original model, one has $\varphi \in \mathrm{deL}[\![\sigma \to \tau]\!]$ if and only if there exists some $f \in \mathcal{M}[\![\sigma \to \sigma]\!]$ such that φ is the equivalence class of f. To verify this is well-defined one then needs to prove that φ induces a map of sets, *i.e.* an element of $\mathbf{Set}(\mathrm{deL}[\![\sigma]\!], \mathrm{deL}[\![\tau]\!])$, and check it has the structure required to be a morphism in the model (*e.g.* Scott continuity / smoothness / …). Our construction avoids this extra work: the required properties on morphisms and function spaces are all imposed by the coreflection H and the fact $\textsc{ohr}(\mathcal{M})$ is a concrete category over $\mathcal{M}$.

Our approach also differs from that taken in games semantics, which has been highly successful in constructing fully abstract models for a variety of languages (*e.g.* Abramsky et al. [1998, 2000]; Clairambault and de Visme [2020]; Hyland and Ong [2000]). In such models the notion of morphism is changed to include more intensional information, and one obtains a fully abstract model by identifying suitable morphisms. Games models are often effectively presentable and may be used to give decision procedures for certain fragments (*e.g.* [Murawski and Tzevelekos 2012]). Our construction, by contrast, is highly non-effective.

We are not aware of any previous fully abstract model for a Moggi-style language with its usual categorical semantics, even for a fixed effect: the closest we know of is the characterisation of contextual equivalence given by Goubault-Larrecq et al. [2004]. Indeed, much of the work in this area has been inspired by PCF. Plotkin [1977] famously showed that, even though the domains model is not fully abstract for PCF, it is fully abstract for PCF extended with parallel-or. This prompted a rich vein of research attempting to classify *sequential* computation, and hence construct a fully abstract model for PCF in domains (see *e.g.* Fiore et al. [1996]). More recent non-games-based constructions generally take their inspiration from this rich literature and focus on languages with recursion: beyond the original O'Hearn–Riecke model, notable examples include Cartwright et al. [1994]; Ehrhard et al. [2014]; Marz [2000]; Matache et al. [2021]; Riecke and Sandholm [2002]. Apart from the first two works, these models employ a similar basic idea to ours, quantifying over a range of 'predictions' which one eventually instantiates to ensure full abstraction. In contrast to our construction, however, all these works construct a category tailored to the language being studied: while they each consider a rich and subtle language, their strategy does not have the range of examples of our λ_c^+-based approach.

1.4 Future Work

Given the subtleties already evident in models over **Set**, we believe the conditions we require on the starting λ_c^+-model to be reasonable. But this work is not the end of the story. Further work is required to cover examples such as presheaf models, or domain-theoretic models of recursion.

We see two immediate next steps. First, loosening the requirement that the original model be well-pointed. We expect that, subject to natural assumptions on the monad and its underlying category, the definability predicate and contextual equivalence predicate (*cf.* Lasota et al. [2007]; Power and Robinson [2000]) should both satisfy logical relations conditions, so that full abstraction at ground types lifts to all higher types. Second, generalising the construction of the hull functor, perhaps using the *comprehension categories* or *subset types* of Jacobs [1993, 1999]. These extensions may enable us to deal with more advanced models, such as presheaf models for local state.

Finally, we would like to extend this work to encompass recursion. First, we would need to show the argument enriches (for example, over $\boldsymbol{\omega}$**Cpo**). Second, we would want to incorporate a notion of approximation to exploit the fact that, in a domain-based model, one may approximate the definable elements from below using suitable compactness properties.

2 EXECUTIVE SUMMARY

In this section we give a high-level overview of our OHR construction. As indicated in Sec. 1.1.3, this broadly follows the steps used to construct $\mathbb{C}_{\mathbf{Fin}}$; the construction is summarised in Figure 1b. We assume a language given by choosing a *signature* S of base types, primitives and effectful operations, together with a *semantic model* consisting of a cartesian closed category $\mathcal{M}$, a (strong) monad T, and a semantic interpretation of the base types and constants. The cartesian closed structure and monad then determine a semantic interpretation $[\![-]\!]$ of all the terms in the language. In this overview we omit sum types as the construction with them is a little more complex.

2.1 Strengthening the Notion of Relation

We saw in Sec. 1.1.2 that in order to construct a fully abstract semantic model it suffices to construct one that is fully complete and well-pointed. As with $\mathbb{C}_{\mathbf{Fin}}$, we shall handle well-pointedness by restricting to a subcategory of concrete objects. Our main aim, therefore, is to soup up the notion of 'relation' in $\mathbb{L}_{\mathbf{Fin}}$ so that every morphism is definable. Our approach is based on two old insights.

Insight 1: Kripke Relations of Varying Arity. Jung and Tiuryn [1993] observed that, because of the variable binding in λ-abstraction, it is highly non-trivial to characterise the definable morphisms in a model of the simply-typed lambda calculus using n-ary relations for a fixed n. They therefore introduced *Kripke relations of varying arity*: a Kripke relation of varying arity on a set $\underline{X}$ consists of a relation $R(\Gamma) \subseteq (\llbracket \Gamma \rrbracket \Rightarrow \underline{X})$ for every context Γ, compatible with variable renaming and weakening. For brevity we call these simply *Kripke relations*.

Kripke relations and relation-preserving morphisms form a category $\mathrm{Krip}_{\mathcal{M},\llbracket - \rrbracket}$ (we shall give a precise definition momentarily). This may be thought of as a version of $\mathbb{L}_{\mathbf{Fin}}$ in which the binary relations R_0 and R_1 are replaced by a family of relations with arities given by the contexts. In other words, the relations are replaced by a *presheaf* on a *category of contexts* (*cf.* [Fiore et al. 1999]).

Definition 2.1. A *context* Γ is a finitely-supported partial map $\mathrm{Var} \rightharpoonup_{\mathrm{fin}} \mathbf{Ty}$ from a countably infinite set of variables to the set of types; we write $\diamond$ for the empty context and $(x : \sigma) \in \Gamma$ for $\Gamma(x) = \sigma$. *Context renamings* $\rho : \Gamma \to \Delta$ are maps $\rho : \mathrm{Dom}\,\Gamma \to \mathrm{Dom}\,\Delta$ that respect the types: if $(x : \sigma) \in \Gamma$ then $(\rho x : \sigma) \in \Delta$. We write $\mathrm{Con}_{\mathrm{S}}$ for the category of contexts and context renamings.

The interpretation $\llbracket - \rrbracket$ defines a functor $\mathrm{Con}_{\mathrm{S}}^{\mathrm{op}} \to \mathcal{M}$: if $\Gamma := (x_i : \sigma_i)_{i=1,\ldots,n}$, $\rho : \Gamma \to \Delta$ and $\rho(x_i) = y_{\tilde{\rho}i}$ for all i then $\llbracket \rho \rrbracket := \langle \pi_{\tilde{\rho}1}, \ldots, \pi_{\tilde{\rho}n} \rangle : \llbracket \Delta \rrbracket \to \llbracket \Gamma \rrbracket$. Classically, Kripke relations of varying arity are defined with respect to this functor. For our purposes later we give a slightly more general definition, in which $\llbracket - \rrbracket$ is replaced by an arbitrary functor and $\mathrm{Con}_{\mathrm{S}}^{\mathrm{op}}$ by an arbitrary category.

Definition 2.2. Let $F : \mathbb{A} \to \mathcal{M}$ be a functor from a small category $\mathbb{A}$. A *Kripke relation of varying arity with respect to* F, which we call simply a *Kripke relation*, on $\underline{X} \in \mathcal{M}$ is a family of predicates $\left\{R(\Gamma) \subseteq \mathcal{M}(F(\Gamma), \underline{X})\right\}_{\Gamma \in \mathbb{A}}$ satisfying the *monotonicity* condition: if $h \in R(\Gamma)$ and $\rho : \Delta \to \Gamma$ is a morphism in $\mathbb{A}$, then $h \circ F\rho \in R(\Delta)$. Kripke relations form a category $\mathrm{Krip}_{\mathcal{M},F}$ with objects $(\underline{X}, R)$ where $\underline{X} \in \mathcal{M}$ and R is a Kripke relation on $\underline{X}$. A morphism $f : (\underline{X}, R) \to (\underline{Y}, S)$ is a map $f : \underline{X} \to \underline{Y}$ in $\mathcal{M}$ that preserves the relation: if $h \in R(\Gamma)$ then $f \circ h \in S(\Gamma)$.

For $F := \llbracket - \rrbracket$ and $\mathcal{M} \subseteq \mathbf{Set}$ the monotonicity condition says that if $\lambda \gamma \,.\, x_\gamma \in R(\Gamma) \subseteq (\llbracket \Gamma \rrbracket \Rightarrow \underline{X})$ and $\rho : \Gamma \to \Delta$ is a context renaming, then $\lambda \delta \,.\, x_{\llbracket \rho \rrbracket \delta} \in R(\Delta) \subseteq (\llbracket \Delta \rrbracket \Rightarrow \underline{X})$. In this setting, $f : (\underline{X}, R) \to (\underline{Y}, S)$ if and only if $\lambda \gamma \,.\, x_\gamma \in R(\Gamma)$ implies $\lambda \gamma \,.\, f(x_\gamma) \in S(\Gamma)$ for every context Γ.

We now make the analogy with $\mathbb{L}_{\mathbf{Fin}}$ precise by showing how $\mathrm{Krip}_{\mathcal{M},F}$ can be constructed in the same way as $\mathbb{L}_{\mathbf{Fin}}$ (recall Figure 1a). As before it is not necessary to follow all the details: our aim is to demonstrate that the same construction is at work. The main step is replacing the fibration for logical relations $\mathrm{cod} : \mathrm{Sub}(\mathbf{Fin}) \to \mathbf{Fin}$ with its indexed counterpart. We therefore replace sub*sets* by sub*presheaves*: a *sub-presheaf* R of $P : \mathbb{A} \to \mathbf{Set}$ is a family $\{R(\Gamma) \subseteq P(\Gamma)\}_{\Gamma \in \mathbb{A}}$ compatible with P's action on morphisms. The $\mathbb{A}$-parametrised version of the subobject fibration employed in Figure 1a is the subobject fibration $\mathrm{cod} : \mathrm{Sub}(\widehat{\mathbb{A}}) \to \widehat{\mathbb{A}}$ taking a sub-presheaf $R \hookrightarrow P$ to P. The category $\mathrm{Krip}_{\mathcal{M},F}$ then arises as the pullback (change-of-base) along the *nerve functor* $\mathrm{N}_F : X \mapsto \mathcal{M}(F(-), X)$, as in Figure 2.

$$\begin{tikzcd}
\mathrm{Krip}_{\mathcal{M},F} \arrow[loop left, "\hat{T}"] \arrow[r] \arrow[d, "\mathrm{U}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Sub}(\widehat{\mathbb{A}}) \arrow[d, "\mathrm{cod}"] \\
\mathcal{M} \arrow[loop left, "T"] \arrow[r, "\mathrm{N}_F"'] & \widehat{\mathbb{A}}
\end{tikzcd}$$

Fig. 2. The construction of $\mathrm{Krip}_{\mathcal{M},F}$

As for $\mathbb{L}_{\mathbf{Fin}}$, it follows from the theory of logical relations of varying arity that $\mathrm{Krip}_{\mathcal{M},F}$ is a cartesian closed category, and that the forgetful functor U strictly preserves this structure (see Sec. 4.1). Moreover, we can use the ⊤⊤*-lifting* of Katsumata [2005] to define a monad $\hat{T}$ on $\mathrm{Krip}_{\mathcal{M},F}$ so that U strictly preserves the monadic structure: $\hat{T}$ is called a *lifting* of T (Def. 4.2). Thus, $\mathrm{Krip}_{\mathcal{M},F}$ is a generalised category of relations and it has has enough structure to interpret the language of interest. The question of which interpretation to take is the subject of our next insight.

Insight 2: Logical Relations. The second insight, due to Plotkin [1973], is that a morphism is definable only if it *satisfies every logical relation*: this is often referred to as the "fundamental lemma" of logical relations. Classically, a logical relation is a family of relations $\{R_\sigma\}_{\sigma\in\mathbf{Ty}}$ indexed by types which is compatible with the type formation rules: for example, one requires that $(h, h') \in R_{\sigma\to\tau}$ if and only if $(x, x') \in R_\sigma$ implies $(h\,x, h'\,x') \in R_\tau$. The definition in our setting is similar, except one must also take account of variable renamings and the monadic effect. For us, a logical relation R consists of a Kripke relation R_σ on $T\llbracket\sigma\rrbracket$ for every type σ, compatible with the type formation rules (Def. 5.1). Thus, R is a family of sets indexed by both types and contexts.

Let us make this more precise. To handle the presence of the effect we use the following operation restricting a Kripke relation to *values*, namely those maps that factor through the monadic unit.

Definition 2.3. For $(T\underline{X}, R) \in \mathrm{Krip}_{\mathcal{M},F}$, define $R^{\mathrm{val}}(\Gamma) := \left\{h : F(\Gamma) \to \underline{X} \,\middle|\, \eta_X \circ h \in R(\Gamma)\right\}$.

Following the strategy for the simply-typed lambda calculus (*e.g.* Alimohamed [1995]; Ma and Reynolds [1992]; Mitchell and Scedrov [1993]), we express compatibility with type formation by compatibility with the structure of $\mathrm{Krip}_{\mathcal{M},\llbracket-\rrbracket}$. For a fixed lifting $\hat{T}$ of T, a *logical relation* is a family of Kripke relations $\{R_\sigma\}_{\sigma\in\mathbf{Ty}}$ satisfying equations such as $(\llbracket\sigma\rrbracket, R^{\mathrm{val}}_\sigma) \Rightarrow (T\llbracket\tau\rrbracket, R_\tau) = (\llbracket\sigma\to\tau\rrbracket, R^{\mathrm{val}}_{\sigma\to\tau})$ and $\hat{T}(\llbracket\sigma\rrbracket, R^{\mathrm{val}}_\sigma) = (T\llbracket\sigma\rrbracket, R_\sigma)$ in $\mathrm{Krip}_{\mathcal{M},\llbracket-\rrbracket}$. This captures exactly the required logical relations conditions. For example, if $\mathcal{M} \subseteq \mathbf{Set}$ the exponential $(\underline{X}, R) \Rightarrow (\underline{Y}, S)$ is $(\underline{X} \Rightarrow \underline{Y}, R \supset S)$, where $\lambda\gamma\,.\,h_\gamma \in (R \supset S)(\Gamma)$ if and only if for all renamings $\rho : \Gamma \to \Delta$ and $\lambda\delta\,.\,x_\delta \in R(\Delta)$ one has $\lambda\delta\,.\,h_{\llbracket\rho\rrbracket(\delta)}(x_\delta) \in S(\Delta)$. Thus one recovers a renaming-compatible version of the classic condition.

Example 2.1. (1) The crucial example of a logical relation is the *definability predicate* given by $\textsc{def}_\sigma(\Gamma) := \left\{\llbracket\Gamma \vdash M : \sigma\rrbracket \mid M \text{ is derivable}\right\}$. Then $f \in \textsc{def}^{\mathrm{val}}_\sigma(\Gamma)$ if and only if $\eta_{\llbracket\sigma\rrbracket} \circ f$ denotes a term. (2) The forgetful functor $\mathrm{U} : \mathbb{L}_{\mathbf{Fin}} \to \mathbf{Fin}$ induces a logical relation $\mathcal{U}$ defined by $\mathcal{U}_\sigma(\Gamma) := \mathbb{L}_{\mathbf{Fin}}(l\llbracket\Gamma\rrbracket, \hat{\mathrm{R}}l\llbracket\sigma\rrbracket) \subseteq \mathbf{Fin}(s\llbracket\Gamma\rrbracket, \mathrm{R}s\llbracket\sigma\rrbracket)$. This phenomenon holds generally: see Prop. 5.1.

A logical relation $R := \{R_\sigma\}_{\sigma\in\mathbf{Ty}}$ determines a semantic interpretation in $\mathrm{Krip}_{\mathcal{M},\llbracket-\rrbracket}$. One sets $l^R\llbracket\beta\rrbracket := (\llbracket\beta\rrbracket, R^{\mathrm{val}}_\beta)$ and observes the equations making R logical entail that $l^R\llbracket\sigma\rrbracket := (\llbracket\sigma\rrbracket, R^{\mathrm{val}}_\sigma)$ and $\hat{T}(l^R\llbracket\sigma\rrbracket) := (T\llbracket\sigma\rrbracket, R_\sigma)$ for every type σ. Following Alimohamed [1995], we say $f : \llbracket\Gamma\rrbracket \to T\llbracket\sigma\rrbracket$ in $\mathcal{M}$ *satisfies* R if and only if $f : l^R\llbracket\Gamma\rrbracket \to \hat{T}(l^R\llbracket\sigma\rrbracket)$ in $\mathrm{Krip}_{\mathcal{M},\llbracket-\rrbracket}$. One then recovers a version of the fundamental lemma (Thm. 5.1) stating that a morphism is definable if and only if it satisfies every logical relation. If f is definable, so $f = \llbracket M\rrbracket$, it lifts to the map $l^R\llbracket M\rrbracket$ in $\mathrm{Krip}_{\mathcal{M},\llbracket-\rrbracket}$ because $\mathrm{U} : \mathrm{Krip}_{\mathcal{M},\llbracket-\rrbracket} \to \mathcal{M}$ preserves the semantic interpretation. Conversely, if f satisfies every logical relation, it satisfies $\textsc{def}$; since $\mathrm{id}_{\llbracket\Gamma\rrbracket} \in \textsc{def}^{\mathrm{val}}_\Gamma(\Gamma)$, this entails that $f \circ \mathrm{id}_{\llbracket\Gamma\rrbracket} = f \in \textsc{def}_\sigma(\Gamma)$.

Summarising, we have the following: (1) restricting to definable morphisms is restricting to those satisfying every logical relation; (2) the morphisms satisfying a logical relation R are exactly those that lift to morphisms respecting the corresponding semantic interpretation in $\mathrm{Krip}_{\mathcal{M},\llbracket-\rrbracket}$.

From the Two Insights to a Fully Abstract Model. The lesson we take from the preceding is that, in order to obtain full completeness, it suffices to adapt the definition of $\mathrm{Krip}_{\mathcal{M},\llbracket-\rrbracket}$ to build a semantic model C in which morphisms necessarily satisfy every logical relation over that model. There is a risk of circularity here: to construct C we need to refer to logical relations over C, but these can only be defined once we have constructed C. O'Hearn & Riecke's way out of this apparent loop is a kind of impredicativity, which they compare to the use of reducibility candidates for proving strong normalisation of the polymorphic λ-calculus (*e.g.* Girard [1989]).

We take a similar approach. We construct $\mathcal{C}$ so that objects are paired not just with one relation but with a whole indexed family of them. By carefully choosing a monad W, a semantic interpretation $l[\![-]\!]$, and the indexing set $\mathbb{I}$, we can ensure that for any type σ and logical relation $\mathcal{R}$ over $\mathcal{C}$ there is $i_0 \in \mathbb{I}$ so that the relation at index i_0 for $l[\![\sigma]\!]$ is exactly $\mathcal{R}^{\mathrm{val}}_{\sigma}$. Morphisms in $\mathcal{C}$ will be morphisms in $\mathcal{M}$ which preserve the relations at every $i \in \mathbb{I}$, so it will follow that any $\mathcal{C}$-morphism must satisfy $\mathcal{R}$. Crucially, if we choose $\mathbb{I}$ large enough, we can ensure i_0 exists using only data over $\mathcal{M}$.

2.2 Constructing the Fully Abstract Model

We start by defining a category $\mathcal{K}$ in which objects are paired with a family of relations indexed by $\mathbb{I}$. For now we leave $\mathbb{I}$ as an arbitrary set; once we have seen the properties we require for full completeness, we shall show how to choose it concretely. To this end we suppose for now that for each $i \in \mathbb{I}$ we have a category $\mathbb{A}_i$ and a functor $F_i : \mathbb{A}_i \to \mathcal{M}$. The objects of $\mathcal{K}$ then consist of an object $\underline{X} \in \mathcal{M}$ together with a Kripke relation $\overline{X}(i)$ for each $i \in \mathbb{I}$, so that $\big(\underline{X}, \overline{X}(i)\big) \in \mathrm{Krip}_{\mathcal{M},F_i}$; thus, each $\overline{X}(i)$ is a family of predicates $\big\{\overline{X}(i)(\Gamma) \subseteq \mathcal{M}(F_i(\Gamma), \underline{X})\big\}_{\Gamma \in \mathbb{A}_i}$ compatible with the action of F_i on morphisms. Morphisms $f : (\underline{X}, \overline{X}) \to (\underline{Y}, \overline{Y})$ in $\mathcal{K}$ are morphisms $f : \underline{X} \to \underline{Y}$ in $\mathcal{M}$ which preserve every relation: if $h \in \overline{X}(i)(\Gamma)$ then $f \circ h \in \overline{Y}(i)(\Gamma)$.

Once again this category arises from the theory of fibrations of logical relations. Fibrations for logical relations are closed under small products, so we can take $\prod_{i \in \mathbb{I}} \mathrm{cod} : \prod_{i \in \mathbb{I}} \mathrm{Sub}(\widehat{\mathbb{A}_i}) \to \prod_{i \in \mathbb{I}} \widehat{\mathbb{A}_i}$ and pull back along $\langle \mathrm{N}_{F_i} \rangle_{i \in \mathbb{I}}$ to obtain $\mathcal{K}$ as in Figure 3. As with $\mathrm{Krip}_{\mathcal{M},F}$, this category is cartesian closed: its structure is given component-wise, and we also define a monad $\hat{T}$ on $\mathcal{K}$ component-wise. The forgetful functor $\mathrm{U} : \mathcal{K} \to \mathcal{M}$ then preserves both the cartesian closed and monadic structure.

$$\begin{tikzcd}
\hat{T} \arrow[loop left] \mathcal{K} \arrow[r] \arrow[d, "\mathrm{U}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \prod_{i \in \mathbb{I}} \mathrm{Sub}(\widehat{\mathbb{A}_i}) \arrow[d, "\prod_{i \in \mathbb{I}} \mathrm{cod}"] \\
T \arrow[loop left] \mathcal{M} \arrow[r, "\langle \mathrm{N}_{F_i} \rangle_{i \in \mathbb{I}}"'] & \prod_{i \in \mathbb{I}} \widehat{\mathbb{A}_i}
\end{tikzcd}$$

Fig. 3. The construction of $\mathcal{K}$

The category $\mathcal{C}$ of our fully abstract model will be the subcategory $j : \mathcal{C} \hookrightarrow \mathcal{K}$ consisting of just the concrete objects, namely those $(\underline{X}, \overline{X}) \in \mathcal{K}$ such that every global element $x : 1 \to \underline{X}$ lifts to a global element $x : 1 \to X$ in $\mathcal{K}$ (*cf.* Def. 1.1). This is always a reflective subcategory and, in the examples we consider, also a coreflective subcategory. Thus $\mathcal{C}$ is cartesian closed and acquires a monad W with underlying functor $\mathrm{H}\hat{T}j$, in the same way as $\mathbb{C}_{\mathbf{Fin}}$. This is summarised in Figure 1b. As for $\mathbb{C}_{\mathbf{Fin}}$, the function space $(X \Rightarrow_{\mathcal{C}} Y)$ in $\mathcal{C}$ has carrier isomorphic to $\mathcal{K}(jX, jY)$ (see Lemma 6.2), so the function space only morphisms preserving the required relations.

It remains to choose $\mathbb{I}$ and the semantic interpretation $l[\![-]\!]$. Following O'Hearn & Riecke, we choose $\mathbb{I}$ in such a way that it necessarily contains the data we need. To simplify applications, we intentionally choose $\mathbb{I}$ to be as large as possible: the more permissive we are in the choice of $\mathbb{I}$, the more properties we have for reasoning within the OHR model.

It turns out that it suffices to choose $\mathbb{I}$ and $l[\![-]\!]$ so that for any logical relation $\mathcal{R} = \{\mathcal{R}_\sigma\}_{\sigma \in \mathbf{Ty}}$ over $\mathcal{C}$ there exists $i_0 \in \mathbb{I}$ so that the relation at i_0 of the interpretation of a base type β is precisely $\mathcal{R}^{\mathrm{val}}_{\beta}$, that is, $\overline{l[\![\beta]\!]}(i_0) = \mathcal{R}^{\mathrm{val}}_{\beta}$. Let us show why this is the case. If $\overline{l[\![\beta]\!]}(i_0) = \mathcal{R}^{\mathrm{val}}_{\beta}$ for every base type β, the cartesian closed structure of $\mathcal{C}$ and the monad W yield $\overline{l[\![\sigma]\!]}(i_0) = \mathcal{R}^{\mathrm{val}}_{\sigma}$ and $\overline{\mathrm{W}(l[\![\sigma]\!])}(i_0) = \mathcal{R}_{\sigma}$ for every type σ (Prop. 7.1). Morphisms in $\mathcal{C}$ preserve the relations at every index, so if $f : l[\![\Gamma]\!] \to \mathrm{W}(l[\![\sigma]\!])$ in $\mathcal{C}$ then $f : \big(\underline{l[\![\Gamma]\!]}, \overline{l[\![\Gamma]\!]}(i)\big) \to \mathrm{W}\big(\underline{l[\![\Gamma]\!]}, \overline{l[\![\Gamma]\!]}(i)\big)$ in $\mathrm{Krip}_{\mathcal{M},F_i}$ for every $i \in \mathbb{I}$; instantiating at i_0 yields $f : \big(\underline{l[\![\Gamma]\!]}, \mathcal{R}^{\mathrm{val}}_{\Gamma}\big) \to \big(\underline{\mathrm{W}(l[\![\Gamma]\!])}, \mathcal{R}_{\sigma}\big)$ in $\mathrm{Krip}_{\mathcal{M},F_{i_0}}$, so f satisfies $\mathcal{R}$.

Now we turn to choosing $\mathbb{I}$. We take it to be a dependent product over enough sets that we can always find the required i_0 (see Sec. 8). An element $i \in \mathbb{I}$ is therefore a tuple $(\mathbb{A}, F, r, \hat{T})$ in which: (1) $\mathbb{A}$ is a category and F is a functor $\mathbb{A} \to \mathcal{M}$; (2) r is a map assigning a concrete Kripke relation over F on $[\![\beta]\!]$ to every base type β, so that $([\![\beta]\!], r(\beta)) \in \mathrm{Krip}_{\mathcal{M},F}$, together with a suitable interpretation of constants; (3) $\hat{T}$ is a lifting of T to $\mathrm{Krip}_{\mathcal{M},F}$, so that the forgetful functor preserves the monadic structure. Note that none of this data refers to $\mathcal{C}$, only to $\mathcal{M}$ or data earlier in the tuple.

To see why this definition works, let $\mathcal{R} := \left\{\mathcal{R}_\sigma(\Gamma) \subseteq \mathcal{C}(l[\![\Gamma]\!], \mathrm{W}l[\![\sigma]\!]\right\}_{\Gamma \in \mathrm{Con}, \sigma \in \mathbf{Ty}}$ be a logical relation over $\mathcal{C}$. For technical reasons we need to assume it satisfies certain *compatibility* and properties (*e.g.* Def. 5.3), but these are mild. We set $i_0 := \left(\mathrm{Con}_{\mathrm{S}}^{\mathrm{op}}, \mathrm{U} \circ l[\![-]\!], r, \hat{T}^{\mathcal{R}}\right)$, where $l[\![-]\!]$ is the semantic interpretation in $\mathcal{C}$ defined below, U is the forgetful functor $\mathcal{C} \longrightarrow \mathcal{M}$, $r : \beta \mapsto \mathcal{R}_\beta^{\mathrm{val}}$, and $\hat{T}^{\mathcal{R}}$ is a monad we construct so that $\hat{T}^{\mathcal{R}}(\mathcal{R}_\sigma^{\mathrm{val}})$ is closely related to $\mathcal{R}_\sigma$ (Lemma 7.3).

We verify first that the the choice of r and the equality $\overline{l[\![\sigma]\!]}(i_0) = \mathcal{R}_\sigma^{\mathrm{val}}$ are well-typed. Via the inclusion $\mathcal{C}(l[\![\Gamma]\!], X) \subseteq \mathcal{M}(\underline{l[\![\Gamma]\!]}, \underline{X})$ we get that $\mathcal{R}$ determines a $\mathbf{Ty}$-indexed family of Kripke relations on $\mathcal{M}$ over the functor $\mathrm{U} \circ l[\![-]\!]$ as follows: $\mathcal{R}_\sigma(\Gamma) \subseteq \mathcal{C}\big(l[\![\Gamma]\!], \mathrm{W}l[\![\sigma]\!]\big) \subseteq \mathcal{M}\big(\mathrm{U}l[\![\Gamma]\!], \underline{\mathrm{W}l[\![\sigma]\!]}\big)$. Thus, we may identify $\mathcal{R}_\sigma$ with an object $(\underline{\mathrm{W}l[\![\sigma]\!]}, \mathcal{R}_\sigma) \in \mathrm{Krip}_{\mathcal{M},\mathrm{U}\circ l[\![-]\!]}$ and $\mathcal{R}_\beta^{\mathrm{val}}$ is a Kripke relation on $\underline{l[\![\beta]\!]}$. So long as the carrier $\underline{l[\![\beta]\!]}$ equals $[\![\beta]\!]$, therefore, the equalities are indeed well-typed.

Finally we show how to guarantee that $\overline{l[\![\beta]\!]}(i_0) = \mathcal{R}_\beta^{\mathrm{val}}$. We do this by choosing the right semantic interpretation. We have just seen that we must set $\underline{l[\![\beta]\!]} = [\![\beta]\!]$ on carriers. On relations, we read off the choice of relation given by the corresponding index: $\overline{l[\![\beta]\!]}(\mathbb{A}, F, r, \hat{T}) := r(\beta)$. In particular, for i_0 we get that $\overline{l[\![\beta]\!]}(i_0) = \mathcal{R}_\beta^{\mathrm{val}}$. As outlined above, it follows that every morphism $l[\![\Gamma]\!] \longrightarrow \mathrm{W}(l[\![\sigma]\!])$ in $\mathcal{C}$ satisfies $\mathcal{R}$, hence is definable: the semantic model given by $\mathcal{C}$, $l[\![-]\!]$ and W is fully complete. Moreover, by our restriction to concrete objects, it is well-pointed. Thus, the model is fully abstract.

2.3 Properties of the OHR Model

We finish this summary by noting some properties of $\mathcal{C}$. Morphisms in $\mathcal{C}$ are morphisms in $\mathcal{M}$ which satisfy every compatible logical relation over $\mathcal{C}$. To show a $\mathcal{M}$-morphism f is not a map in $\mathcal{C}$, therefore, it suffices to find a logical relation over $\mathcal{C}$ which f does not satisfy. In the presence of effects, doing this directly this can be difficult. We mitigate this in two ways. First, we show that logical relations can be constructed by constructing a semantic model (Prop. 5.1). Often this can be done using an intuitive relational condition, as in the construction of $\mathbb{L}_{\mathbf{Fin}}$ in Ex. 1.1. Second, we indicate how one may induce logical relations on $\mathcal{C}$ using certain logical relations on $\mathcal{M}$ (Sec. 9).

The semantic interpretation of ground types in $\mathcal{C}$ coincides with that in $\mathcal{M}$. Moreover, in our examples the monad W on $\mathcal{C}$ is a sub-monad of the monad T on $\mathcal{M}$: there is a canonical monad morphism giving a monic $\mathrm{c}_X : \underline{\mathrm{W}X} \hookrightarrow T\underline{X}$ for every $X \in \mathcal{C}$ (Lemma 6.1). Hence, the interpretations of closed ground terms in $\mathcal{C}$ are determined by those in $\mathcal{M}$ (Lemma 8.2). This disappears at higher types because the forgetful functor $\mathcal{C} \longrightarrow \mathcal{M}$ does not preserve function spaces (*cf.* Remark 1.1).

It follows that $\mathcal{C}$ inherits much of the intuition of the original model. For example, in the OHR model over the model $\mathbb{L}_{\mathbf{Fin}}$ of Ex. 1.1, the maps are set maps, the carriers of the function spaces are subsets of those in $\mathbf{Fin}$, and the carrier of $\mathrm{W}X$ is a subset of $\mathrm{R}\underline{X}$. We examine this further in Sec. 9.

Finally, we do not yet incorporate recursion so our model does not use any form of approximation. While O'Hearn & Riecke use a form of compactness to approximate every term using a countable chain, no such facility is available to us. We hope to pursue this line in the future (see Sec. 1.4).

3 THE COMPUTATIONAL λ-CALCULUS λ_c^+

Syntax. Throughout we shall assume a fixed (but arbitrary) choice of λ_c^+-*signature* S, consisting of

(1) A set $\mathbf{B}$ of *base types*, which we extend with a unit type, binary products, an empty type and binary sums to obtain the set of *ground types*: $\mathbf{G} \ni \gamma ::= \beta \in \mathbf{B} \mid 1 \mid \gamma * \gamma \mid 0 \mid \gamma + \gamma$. Including function types yields the set of *simple types*: $\mathbf{Ty}^+ \ni \sigma ::= \beta \in \mathbf{B} \mid 1 \mid \sigma * \sigma \mid 0 \mid \sigma + \sigma \mid \sigma \longrightarrow \sigma$.

(2) A set $\mathbf{E}$ of (algebraic) *effect operations* [Plotkin and Power 2003] and an assignment $\mathbf{E} \longrightarrow \mathbf{G}\times\mathbf{G}$ assigning a *parameter type* α and an *arity type* κ to every $\mathrm{op} \in \mathbf{E}$.
(3) A set $\mathbf{P}$ of *primitives* and an assignment $\mathbf{P} \longrightarrow \mathbf{Ty}^{+}$ of a simple type κ to every $\xi \in \mathbf{P}$. We require that either $\kappa \in \mathbf{G}$ or $\kappa = \sigma_1 * \cdots * \sigma_n \to \gamma$ where $\gamma \in \mathbf{G}$ and each σ_i is a (possibly *thunked*) ground type: either $\sigma_i \in \mathbf{G}$, or $\sigma_i = (1 \to \gamma_i)$ and $\gamma_i \in \mathbf{G}$.

We write $\mathrm{op} : \alpha \rightsquigarrow \kappa$ to indicate that $\mathrm{op} \in \mathbf{E}$ has parameter type α and arity type κ, and $\xi : \kappa$ to indicate that $\xi \in \mathbf{P}$ is assigned type κ. Although we do not take advantage of it in this paper, we distinguish between effect operations and primitives so that future developments can employ the better metatheory enjoyed by effect operations (*e.g.* Kammar and Plotkin [2012]; Katsumata [2013]).

Example 3.1. For a set of exceptions E consider the *exception monad* $(-) + E$ together with an effect operation raise_e raising an exception for each $e \in E$. The corresponding handle_e operation is not algebraic ([Plotkin and Power 2003, Example 3]), but one could add it as a primitive.

The typing rules of the type theory $\lambda_c^{+}(\mathrm{S})$ are the usual rules for the simply-typed lambda calculus with finite products and finite sum types (*e.g.* Fiore and Simpson [1999]; Scherer [2017]) together with the rules for effect operations and primitives:

$$\frac{\Gamma \vdash M : \alpha}{\Gamma \vdash \mathrm{op}\, M : \kappa}\ (\mathrm{op} : \alpha \rightsquigarrow \kappa) \qquad\qquad \frac{}{\Gamma \vdash \xi : \kappa}\ (\xi : \kappa)$$

Semantic Interpretation. A $\lambda_c^{+}(\mathrm{S})$*-model* $(\mathcal{M}, T, s)$ consists of

(1) A category $\mathcal{M}$ with finite products $(\times, 1)$, finite coproducts $(+, 0)$ and exponentials $\Rightarrow$ (that is, a *bi-cartesian closed category* or *bi-CCC*).
(2) A functor $T : \mathcal{M} \longrightarrow \mathcal{M}$ equipped with natural transformations with components $\mu_A : TTA \longrightarrow TA$, $\eta_A : A \longrightarrow TA$, and $\mathrm{st}_{A,B} : A \times TB \longrightarrow T(A \times B)$ satisfying standard axioms [Kock 1972] (that is, a *strong monad* $(T, \mu, \eta, \mathrm{st})$). Where the context is unambiguous we write simply T.
(3) An interpretation $s : \mathbf{B} \longrightarrow \mathrm{Ob}(\mathcal{M})$ of base types. This extends to an interpretation $s[\![-]\!] : \mathbf{Ty}^{+} \longrightarrow \mathrm{Ob}(\mathcal{M})$ of all types; on contexts we set $s[\![\Gamma]\!] := \prod_{(x:\sigma)\in\Gamma} s[\![\sigma]\!]$.
(4) An interpretation of effect operations and primitives: a Kleisli arrow $s[\![\mathrm{op}]\!] : s[\![\alpha]\!] \longrightarrow Ts[\![\kappa]\!]$ for every $\mathrm{op} : \alpha \rightsquigarrow \kappa$ and a global element $s[\![\xi]\!] : 1 \longrightarrow s[\![\kappa]\!]$ for every $\xi : \kappa$.

A *morphism of* $\lambda_c^{+}(\mathrm{S})$*-models* $F : (\mathcal{M}, T, s) \longrightarrow (\mathcal{M}', T', s')$ is a functor $F : \mathcal{M} \longrightarrow \mathcal{M}'$ which strictly preserves all the structure. Thus, F must strictly preserve the cartesian closed structure and satisfy $FT = T'F$, $F\mu_X = \mu'_{FX}$, $F\eta_X = \eta'_{FX}$, $F\,\mathrm{st}_{X,Y} = \mathrm{st}'_{FX,FY}$ and $Fs = s'$ for all $X \in \mathcal{M}$.

The interpretation s in a $\lambda_c^{+}(\mathrm{S})$-model $(\mathcal{M}, T, s)$ extends to an interpretation $s[\![-]\!]$ of all $\lambda_c^{+}(\mathrm{S})$-terms: $s[\![\Gamma \vdash M : \tau]\!] : s[\![\Gamma]\!] \longrightarrow T(s[\![\tau]\!])$. The rules are standard (*e.g.* Fiore and Simpson [1999]; Moggi [1989]) so we omit them. A $\lambda_c^{+}(\mathrm{S})$-model morphism strictly preserves these interpretations: if $F : (\mathcal{M}, T, s) \longrightarrow (\mathcal{M}', T', s')$ and $(\Gamma \vdash M : \sigma)$ then $F(s[\![\Gamma \vdash M : \sigma]\!]) = s'[\![\Gamma \vdash M : \sigma]\!]$.

We shall also consider the development without sum types. Write $\lambda_c(\mathrm{S})$ for the fragment of $\lambda_c^{+}(\mathrm{S})$ without any sum types, and $\mathbf{Ty}$ for the subset of $\mathbf{Ty}^{+}$ defined by $\mathbf{Ty} \ni \sigma ::= \beta \in \mathbf{B} \mid 1 \mid \sigma * \sigma \mid \sigma \to \sigma$. A $\lambda_c(\mathrm{S})$*-model* consists of a CCC $(\mathcal{M}, \times, 1, \Rightarrow)$ equipped with a strong monad, an interpretation of base types and an interpretation of each operation op and primitive ξ.

We write $f^{\#}$ for the *Kleisli extension* $\mu_Y \circ Tf : TX \longrightarrow TY$ of $f : X \longrightarrow TY$.

Full Abstraction and Full Completeness. We streamline the development by using the *semantic* contextual equivalence of Milner [1977]. We say two $\lambda_c^{+}(\mathrm{S})$-terms $(\Gamma \vdash M : \sigma)$ and $(\Gamma \vdash M' : \sigma)$ are *contextually equivalent* in a $\lambda_c^{+}(\mathrm{S})$-model $(\mathcal{M}, T, s)$, and write $(\Gamma \vdash M \simeq_{\mathrm{ctx}} M' : \sigma)$, if for every closed ground context $C[-]$ one has $s[\![\diamond \vdash C[M] : \gamma]\!] = s[\![\diamond \vdash C[M'] : \gamma]\!]$. The model is *fully abstract* if contextual equivalence implies denotational equality: $s[\![\Gamma \vdash M : \tau]\!] = s[\![\Gamma \vdash M' : \tau]\!]$ whenever $(\Gamma \vdash M \simeq_{\mathrm{ctx}} M' : \sigma)$. This notion coincides with the syntactic definition in common examples, such as the domains model of PCF. The converse to full abstraction—called *adequacy*—follows from the compositionality of the interpretation $s[\![-]\!]$. We now substantiate the claim made in Sec. 1.1.2.

Proposition 3.1. *Any well-pointed, fully complete $\lambda_c^+(\mathrm{S})$-model is fully abstract.*

The proof echoes the *definable separability* criterion of Curien [2007]; one uses well-pointedness together with the fact that for any term $(\Gamma \vdash M : \tau)$ and environment $(\diamond \vdash G : \prod_{(x:\sigma)\in\Gamma} \sigma)$ there is a context $C[-]$ such that $s[\![\Gamma \vdash C[M] : \tau]\!] = s[\![\Gamma \vdash M : \tau]\!]^{\#} \circ s[\![\diamond \vdash G : \prod_{(x:\sigma)\in\Gamma} \sigma]\!]$.

4 THE FIBRATIONAL APPROACH TO DEFINABILITY

In this section we introduce the technical machinery referred to in Sec. 2, namely *fibrations for logical relations*, which axiomatise the data required to study logical relations, $\top\top$-*lifting*, which allows us to construct monads on categories of relations, and $\top\top$-*closure*, for handling sum types.

4.1 Kripke Relations of Varying Arity and Fibrations for Logical Relations

We assume familiarity with the basics of fibrations: see *e.g.* Jacobs [1999] or Loregian and Riehl [2020]. For any fibration $p : \mathbb{E} \to \mathbb{B}$ and product-preserving functor $\mathrm{N} : \mathbb{C} \to \mathbb{B}$, write $\mathrm{U} : \mathrm{Krip}(p, \mathrm{N}) \to \mathbb{C}$ for the fibration obtained by *change-of-base* as in the (pullback) diagram in Figure 4.

Objects of $\mathrm{Krip}(p, \mathrm{N})$ are pairs $(\underline{X} \in \mathbb{C}, R \in \mathbb{E})$ such that $\mathrm{N}\underline{X} = pR$; morphisms are pairs $(f : \underline{X} \to \underline{X'}, \hat{f} : R \to R')$ such that $\mathrm{N}f = p(\hat{f})$. We want to study cases in which $\mathrm{Krip}(p, \mathrm{N})$ is a bi-CCC and U strictly preserves this structure. This is captured by the next definition, due to Katsumata [2013].

$$\begin{tikzcd}
\mathrm{Krip}(p,\mathrm{N}) \arrow[r] \arrow[d, "\mathrm{U}"'] \arrow[dr, phantom, very near start, "\lrcorner"] & \mathbb{E} \arrow[d, "p"] \\
\mathbb{C} \arrow[r, "\mathrm{N}"'] & \mathbb{B}
\end{tikzcd}$$

Fig. 4. Constructing $\mathrm{Krip}(p, \mathrm{N})$

Definition 4.1. A *fibration for logical relations* over a bi-CCC $\mathbb{B}$ is a partial order bifibration $p : \mathbb{E} \to \mathbb{B}$ with small fibres and fibrewise small products such that $\mathbb{E}$ is a bi-CCC and p strictly preserves the bi-CCC structure.

Sec. 1.1.3 employed the category $\mathrm{Sub}(\mathbf{Fin})$ of unary *predicates* on **Fin**; the functor $\mathrm{Sub}(\mathbf{Fin}) \to \mathbf{Fin} : (A, X) \mapsto X$ is a fibration for logical relations. For more examples see [Katsumata 2013, §6].

Lemma 4.1 ([Katsumata 2013, Prop. 6])**.** *For any fibration for logical relations $p : \mathbb{E} \to \mathbb{B}$, bi-CCC $\mathbb{C}$ and product-preserving functor* $\mathrm{N} : \mathbb{C} \to \mathbb{B}$, $\mathrm{U} : \mathrm{Krip}(p, \mathrm{N}) \to \mathbb{C}$ *is a fibration for logical relations.*

As noted above, the categories $\mathbb{L}_{\mathbf{Fin}}$, $\mathrm{Krip}_{\mathcal{M},F}$ and $\mathcal{K}$ all arise as instances of this lemma. In particular, where N_F denotes the nerve functor $X \mapsto \mathcal{M}\big(F(-), X\big)$, we have $\mathrm{Krip}_{\mathcal{M},F} := \mathrm{Krip}(\mathrm{cod}, \mathrm{N}_F)$ and $\mathcal{K} := \mathrm{Krip}(\prod_i \mathrm{cod}, \mathrm{N}_{\langle F_i \rangle_i})$. The bi-CCC structure of $\mathrm{Krip}_{\mathcal{M},F}$ is given below. The exponentials encode the Jung–Tiuryn *logical relations condition* (*cf.* Jung and Tiuryn [1993]; Plotkin [1980]).

Products: The terminal object is $(1_{\mathbb{C}}, \top)$, where $\top(\Gamma) = \{!_{F\Gamma}\}$. The binary product is $(\underline{A}_1, R_1) \times (\underline{A}_2, R_2) := (\underline{A}_1 \times \underline{A}_2, R_1 \circledast R_2)$, where $h \in (R_1 \circledast R_2)(\Gamma)$ if and only if $\pi_i \circ h \in R_i(\Gamma)$ for $i = 1, 2$.

Exponentials: $(\underline{A}, R) \Rightarrow (\underline{B}, S) := (\underline{A} \Rightarrow \underline{B}, R \supset S)$, where $h \in (R \supset S)(\Gamma)$ if and only if $\mathrm{eval} \circ \langle h \circ F\rho, u \rangle \in S(\Delta)$ for every morphism $\rho : \Delta \to \Gamma$ and every $u \in R(\Delta)$.

Coproducts: The initial object is $(0_{\mathbb{C}}, \bot)$, where $\bot(\Gamma) = \emptyset$. The binary coproduct is $(\underline{A}_1, R_1) + (\underline{A}_2, R_2) := (\underline{A}_1 + \underline{A}_2, R_1 \oplus R_2)$, with $(R_1 \oplus R_2)(\Gamma) := \big\{\mathrm{inj}_1 \circ h \,\big|\, h \in R_1(\Gamma)\big\} \cup \big\{\mathrm{inj}_2 \circ h \,\big|\, h \in R_2(\Gamma)\big\}$.

4.2 Monad Liftings and Generalised $\top\top$-Lifting

By Lemma 4.1, if we start from a $\lambda_c^+(\mathrm{S})$-model $(\mathcal{M}, T, s)$ then the category of Kripke relations $\mathrm{Krip}_{\mathcal{M},s[\![-]\!]}$ is a bi-CCC. To make this into a $\lambda_c^+(\mathrm{S})$-model we want a *lifting* of T.

Definition 4.2 ([Katsumata 2005, 2013])**.** Let $p : \mathbb{E} \to \mathbb{B}$ be a fibration for logical relations and $(T, \mu, \eta, \mathrm{st})$ a strong monad on $\mathbb{B}$. A *lifting* of T is a strong monad $(\hat{T}, \hat{\mu}, \hat{\eta}, \hat{\mathrm{st}})$ on $\mathbb{E}$ such that $p\hat{T} = Tp$, $p\hat{\mu} = \mu_p$, $p\hat{\eta} = \eta_p$, and $p\hat{\mathrm{st}} = \mathrm{st}_p$.

Our characterisation of definability will rely on the *semantic* $\top\top$*-lifting* of Katsumata [2005]. Alternative liftings include the *free lifting* [Kammar and McDermott 2018] and the monadic lifting of Goubault-Larrecq et al. [2008]. We choose Katsumata's approach because it interacts well with the extra structure we shall need at sum types (see Def. 5.2).

The next lemma describes the abstract situation.

Lemma 4.2. *Let* $p : \mathbb{E} \to \mathbb{B}$ *be a fibration for logical relations and* T *and* S *be strong monads on* $\mathbb{B}$ *related by a morphism of strong monads* $\alpha : T \Rightarrow S$ *[Pareigis 1977; Street 1972]. Further assume that* S *has a lifting* $\hat{S}$ *and set* $\hat{T}X$ *to be the cartesian lifting below:*

$$\begin{tikzcd} \hat{T}X \arrow[r, dashed, "\hat{\alpha}_X"] & \hat{S}X & & \mathbb{E} \arrow[d, "p"] \\ T(pX) \arrow[r, "\alpha_{pX}"'] & S(pX) & & \mathbb{B} \end{tikzcd} \tag{3}$$

Then $\hat{T}(-)$ *extends to a monad which is a lifting of* T*, and* $\hat{\alpha}$ *extends to a morphism of strong monads.*

$\top\top$*-lifting* arises by taking S and $\hat{S}$ to both be the (strong) continuation monad. In any CCC $\mathbb{C}$, let K_P denote the continuation monad $(- \Rightarrow P) \Rightarrow P$; there exists a canonical monad morphism $\sigma^P : T \Rightarrow \mathrm{K}_{TP}$ with components $\lambda(\mathrm{eval}^{\#} \circ \mathrm{st} \circ \langle \pi_2, \pi_1 \rangle)$. When $p : \mathbb{E} \to \mathbb{B}$ is a fibration for logical relations, write $\hat{\mathrm{K}}_P$ for the continuation monad on $\mathbb{E}$ with target P and observe that this is a lifting of $\mathrm{K}_{p(P)}$. We call an object $P \in \mathbb{E}$ such that $p(P) = TQ$ for some $Q \in \mathbb{B}$ a $\top\top$*-lifting parameter*.

Definition 4.3 ([Katsumata 2005])**.** Let $p : \mathbb{E} \to \mathbb{B}$ be a fibration for logical relations and T be a strong monad on $\mathbb{B}$. Where $p(P) = TQ$ is a $\top\top$-lifting parameter, the $\top\top[P]$*-lifting* $T^{\top\top[P]}$ of T is the monad obtained by taking $\hat{S} := \hat{\mathrm{K}}_P$, $S := \mathrm{K}_{TQ}$ and $\alpha := \sigma^Q$ in (3). For a small set $\mathbb{P} \neq \emptyset$ of $\top\top$-lifting parameters, the $\top\top[\mathbb{P}]$*-lifting of* T is the fibred product $T^{\top\top[\mathbb{P}]} := \bigwedge_{P \in \mathbb{P}} T^{\top\top[P]}(-)$.

We shall implicitly assume that every set of $\top\top$-lifting parameters is both non-empty and small.

4.3 Generalised $\top\top$-Closure

While the definability predicate for a $\lambda_c^+(\mathrm{S})$-model $(\mathcal{M}, T, s)$ interacts well with the cartesian closed structure of $\mathrm{Krip}_{\mathcal{M},s[\![-]\!]}$, this is not the case for the cocartesian structure (see Sec. 5). We therefore follow Katsumata [2008] in restricting to a subcategory with a different coproduct structure.

Definition 4.4. Let $p : \mathbb{E} \to \mathbb{B}$ be a fibration for logical relations, T be a strong monad on $\mathbb{B}$, and $\hat{T}$ be a lifting of T to $\mathbb{E}$. Define a monad $(-)^{\top\top[\hat{T}]}$ lifting the identity monad by taking $\hat{S} := \hat{T}$, $S := T$, $T := \mathrm{id}$ and $\alpha := \eta^T$ in (3). If $X = X^{\top\top[\hat{T}]}$ we say X is $\hat{T}$*-closed.* We write $i : \mathbb{E}^{\top\top[\hat{T}]} \hookrightarrow \mathbb{E}$ for the full subcategory of $\hat{T}$-closed objects.

The $\top\top$*-closure* of Katsumata [2008] arises as a special case. Where $\mathbb{P}$ is a set of $\top\top$-lifting parameters, an object $X \in \mathbb{E}$ is $\top\top[\mathbb{P}]$*-closed*, or simply $\top\top$*-closed*, if $\mathrm{id}^{\top\top[\mathbb{P}]}(X) = X$.

Lemma 4.3. *Let* $p : \mathbb{E} \to \mathbb{B}$ *be a fibration for logical relations and* T *be a strong monad on* $\mathbb{B}$*. For any set* $\mathbb{P}$ *of* $\top\top$*-lifting parameters,* $(-)^{\top\top[T^{\top\top[\mathbb{P}]}]} = \mathrm{id}^{\top\top[\mathbb{P}]}$.

Hence, X is $\top\top[\mathbb{P}]$-closed if and only if it is $T^{\top\top[\mathbb{P}]}$-closed. We finish this section by recording some elementary properties of the subcategory of $\hat{T}$-closed objects. The first two cases generalise properties of the subcategory of $\top\top$-closed objects proven in [Katsumata 2008].

Lemma 4.4. *Let $p : \mathbb{E} \to \mathbb{B}$ be a fibration for logical relations, T be a strong monad on $\mathbb{B}$, and $\hat{T}$ be a lifting of T to $\mathbb{E}$. Then:*

(1) $(-)^{\top\top[\hat{T}]}$ is an idempotent monad, so $\mathbb{E}^{\top\top[\hat{T}]}$ is a replete, reflective subcategory;

(2) $\mathbb{E}^{\top\top[\hat{T}]}$ is a sub-CCC of $\mathbb{E}$, and has finite coproducts as $X_i \xrightarrow{\mathrm{inj}_i} \sum_{i\le n} X_i \xrightarrow{\le} \mathrm{id}^{\top\top[\hat{T}]}\left(\sum_{i\le n} X_i\right)$;

(3) $\hat{T}$ restricts to a strong monad on $\mathbb{E}^{\top\top[\hat{T}]}$: if X is $\hat{T}$-closed, then so is $\hat{T}X$.

5 DEFINABILITY AND $\lambda_c^+(\mathrm{S})$-LOGICAL RELATIONS

In Sec. 2.1 our second key observation was that the the definable morphisms in a fixed $\lambda_c^+(\mathrm{S})$-model $(\mathcal{M}, T, s)$ may be characterised as as those which *satisfy* every *logical relation*, and that this can be expressed in categorical terms: a map $f : s[\![\Gamma]\!] \to T(s[\![\sigma]\!])$ in $\mathcal{M}$ satisfies R if and only if it lifts to a morphism of Kripke relations $f : l^R[\![\Gamma]\!] \to \hat{T}(l^R[\![\sigma]\!])$. We now make this precise.

Our starting point is the next definition, which extends both logical relations for the simply-typed λ-calculus and logical relations for the monadic metalanguage over **Set** [Katsumata 2005]. Since we use $\lambda_c^+(\mathrm{S})$ rather than the monadic metalanguage we use the $(-)^{\mathrm{val}}$ operation (Def. 2.3).

Definition 5.1. Let $\hat{T}$ be a lifting of T to $\mathrm{Krip}_{\mathcal{M},s[\![-]\!]}$. A $\lambda_c(\mathrm{S})$*-logical relation over* $\hat{T}$ is a **Ty**-indexed family $\{R_\sigma \hookrightarrow \mathcal{M}(s[\![-]\!], Ts[\![\sigma]\!])\}_{\sigma\in\mathbf{Ty}}$ such that for every $\sigma, \tau, \sigma_1, \sigma_2 \in \mathbf{Ty}$ we have

$$R_1^{\mathrm{val}} = \top, \qquad R_{\sigma_1 * \sigma_2}^{\mathrm{val}} = R_{\sigma_1}^{\mathrm{val}} \circledast R_{\sigma_2}^{\mathrm{val}}, \qquad R_{\sigma\to\tau}^{\mathrm{val}} = (R_\sigma^{\mathrm{val}} \supset R_\tau), \qquad \hat{T}(s[\![\sigma]\!], R_\sigma^{\mathrm{val}}) = (Ts[\![\sigma]\!], R_\sigma).$$

Moreover, we require that the interpretation of every effect operation $\mathrm{op} : \alpha \rightsquigarrow \kappa$ and primitive $(\xi : \kappa)$ lifts to Kripke relations: $s[\![\mathrm{op}\,]\!] : (s[\![\alpha]\!], R_\alpha^{\mathrm{val}}) \to (Ts[\![\kappa]\!], R_\kappa)$ and $s[\![\xi]\!] : (1, \top) \to (s[\![\kappa]\!], R_\kappa^{\mathrm{val}})$. We extend the definition to contexts using the product structure, so that $R_\Gamma^{\mathrm{val}} := \circledast_{(x:\sigma)\in\Gamma} R_\sigma^{\mathrm{val}}$.

A $\lambda_c(\mathrm{S})$-logical relation is equivalently an interpretation of base types $\hat{s}^R : \beta \mapsto (s[\![\beta]\!], R_\beta^{\mathrm{val}})$ which extends to an interpretation satisfying $\hat{s}^R[\![\sigma]\!] = (s[\![\sigma]\!], R_\sigma^{\mathrm{val}})$ for every $\sigma \in \mathbf{Ty}$.

We cannot use the coproduct structure of $\mathrm{Krip}_{\mathcal{M},s[\![-]\!]}$ to incorporate sum types because elements of $\mathrm{DEF}_{\sigma_1}^{\mathrm{val}} \oplus \mathrm{DEF}_{\sigma_2}^{\mathrm{val}}$ must factor through one of the injections $s[\![\sigma_i]\!] \to s[\![\sigma_1]\!] + s[\![\sigma_2]\!]$, so one has strict inclusions $\bot \subsetneq \mathrm{DEF}_0^{\mathrm{val}}$ and $(\mathrm{DEF}_{\sigma_1}^{\mathrm{val}} \oplus \mathrm{DEF}_{\sigma_2}^{\mathrm{val}}) \subsetneq \mathrm{DEF}_{\sigma_1+\sigma_2}^{\mathrm{val}}$. We therefore restrict to the subcategory of $\hat{T}$-closed objects. Using Lemma 4.4, one obtains the following diagram for any lifting $\hat{T}$ of T. The inclusion i strictly preserves cartesian closed structure but does *not* preserve coproducts.

$$\begin{tikzcd} \mathrm{Krip}^{\top\top[\hat{T}]}_{\mathcal{M},s[\![-]\!]} \arrow[loop left, "\hat{T}"] \arrow[r, hook, bend right, "i"'] \arrow[r, phantom, "\perp"] & \mathrm{Krip}_{\mathcal{M},s[\![-]\!]} \arrow[l, bend right, "(-)^{\top\top[\hat{T}]}"'] \arrow[loop right, "\hat{T}"] \end{tikzcd} \qquad (4)$$

We now define $\lambda_c^+(\mathrm{S})$-logical relations by extending Def. 5.2 with cases for sums; as in Def. 5.2, this extends to contexts using the product structure.

Definition 5.2 (*cf.* [Katsumata 2008, §3.4])**.** Let $\hat{T}$ be a lifting of T to $\mathrm{Krip}_{\mathcal{M},s[\![-]\!]}$. A $\lambda_c^+(\mathrm{S})$*-logical relation* over $\hat{T}$ is a $\mathbf{Ty}^+$-indexed family $\{R_\sigma \hookrightarrow \mathcal{M}(s[\![-]\!], Ts[\![\sigma]\!])\}_{\sigma\in\mathbf{Ty}^+}$ such that: (1) the conditions of Def. 5.1 hold; and (2) $R_0^{\mathrm{val}} = \bot^{\top\top[\hat{T}]}$ and $(R_{\sigma_1}^{\mathrm{val}} \oplus R_{\sigma_2}^{\mathrm{val}})^{\top\top[\hat{T}]} = R_{\sigma_1+\sigma_2}^{\mathrm{val}}$ for every $\sigma_1, \sigma_2 \in \mathbf{Ty}^+$.

Remark 5.1. When $\hat{T}$ is the $\top\top$-lifting $T^{\top\top[\mathbb{P}]}$, so that $R^{\top\top[\hat{T}]} = \mathrm{id}^{\top\top[\mathbb{P}]}R$, it suffices to require that $\big((R_{\sigma_1}^{\mathrm{val}} \oplus R_{\sigma_2}^{\mathrm{val}}) \supset Q\big) = \big(R_{\sigma_1+\sigma_2}^{\mathrm{val}} \supset Q\big)$ and $(R_0^{\mathrm{val}} \supset Q) = (\bot \supset Q)$ for any $Q \in \mathbb{P}$ and $\sigma_1, \sigma_2 \in \mathbf{Ty}^+$: modulo the $(-)^{\mathrm{val}}$ operation, this is the condition in Katsumata's definition.

$\lambda_c^+(\mathrm{S})$-logical relations satisfy the same key property as $\lambda_c(\mathrm{S})$-logical relations.

Lemma 5.1. *Let R be a $\lambda_c^+(\mathrm{S})$-logical relation over $\hat{T}$. Then every $(s[\![\sigma]\!], R_\sigma^{\mathrm{val}})$ and $(Ts[\![\sigma]\!], R_\sigma)$ is $\hat{T}$-closed. Hence if $\hat{s}^R(\beta) = \big(s[\![\beta]\!], R_\beta^{\mathrm{val}}\big)$ for all base types β then $\hat{s}^R[\![\sigma]\!] = \big(s[\![\sigma]\!], R_\sigma^{\mathrm{val}}\big)$ for all $\sigma \in \mathbf{Ty}^+$.*

We now characterise the definable morphisms as those which satisfy every $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-logical relation. To do so, we take a detour through $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-model morphisms. We saw in Ex. 1.2 that such functors exclude 'counterexample' morphisms from a model; it turns out this approach is implicitly using $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-logical relations. As well as being a key step in our characterisation of definability, the next result is useful because it is often more intuitive to construct a model than a $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-logical relation.

Let $F : (\mathcal{M}, T, s) \to (\mathcal{N}, S, t)$ be a morphism of $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-models. Since F strictly preserves all the structure, we obtain $\mathcal{F}_{\sigma}(\Gamma) := \left\{ Fh \,\middle|\, h \in \mathcal{M}(s[\![\Gamma]\!], Ts[\![\sigma]\!]) \right\} \subseteq \mathcal{N}(t[\![\Gamma]\!], St[\![\sigma]\!])$ for every $\sigma \in \mathbf{Ty}^{+}$.

Notation 5.1. Every family $R := \{(Ts[\![\sigma]\!], R_{\sigma})\}_{\sigma \in \mathbf{Ty}^{+}}$ and hence every $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-logical relation, determines a family of $\top\top$-lifting parameters: we also denote this by R.

Proposition 5.1. *For any strict $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-model morphism $F : (\mathcal{M}, T, s) \to (\mathcal{N}, S, t)$, the family $\mathcal{F} := \{\mathcal{F}_{\sigma}\}_{\sigma \in \mathbf{Ty}^{+}}$ defined above is a $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-logical relation over the monad $S^{\top\top[\mathcal{F}]}$ lifting S to* $\mathrm{Krip}_{\mathcal{N}, t[\![-]\!]}$.

Thus, we may recast our construction of the category $\mathbb{L}_{\mathbf{Fin}}$ in Ex. 1.2 as restricting to the subcategory of **Set** in which morphisms preserve the $\lambda_{\mathrm{c}}(\mathrm{S})$-logical relation $\mathcal{U}$ of Ex. 2.1.

Example 5.1. The definability predicate DEF (Ex. 2.1) arises from Prop. 5.1: for a model $(\mathcal{M}, T, s)$ take the subcategory of $\mathcal{M}$ with objects $\{s[\![\sigma]\!]\}_{\sigma \in \mathbf{Ty}^{+}}$ and morphisms the definable maps.

The desired characterisation now follows from Prop. 5.1, Ex. 5.1, and the argument sketched in Sec. 2.1; a similar result holds for $\lambda_{\mathrm{c}}(\mathrm{S})$-relations. Say that a morphism $f : s[\![\Gamma]\!] \to Ts[\![\sigma]\!]$ in $\mathcal{M}$ *satisfies* a $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-logical relation $\{R_{\sigma}\}_{\sigma \in \mathbf{Ty}^{+}}$ if f lifts to a morphism $\hat{s}^{R}[\![\Gamma]\!] \to \hat{T}(\hat{s}^{R}[\![\sigma]\!])$ in $\mathrm{Krip}_{\mathcal{M}, s[\![-]\!]}^{\top\top[\hat{T}]}$.

Theorem 5.1. DEF $:= \{\mathrm{DEF}_{\sigma}\}_{\sigma \in \mathbf{Ty}^{+}}$ *is a $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-logical relation over $T^{\top\top[\mathrm{DEF}]}$. Hence, $f : s[\![\Gamma]\!] \to Ts[\![\sigma]\!]$ in $\mathcal{M}$ is $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-definable if and only if it satisfies every $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-logical relation over $T^{\top\top[\mathrm{DEF}]}$.*

The logical relations arising via Prop. 5.1 are particularly well-behaved; for example, they are closed under currying. We end this section by axiomatising these properties.

Definition 5.3. A $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-logical relation $R = \{R_{\sigma}\}_{\sigma \in \mathbf{Ty}^{+}}$ over $\hat{T}$ is: (1) *hungry* if $\mathrm{id}_{s[\![\sigma]\!]} \in R_{\sigma}^{\mathrm{val}}(x : \sigma)$ for all $\sigma \in \mathbf{Ty}^{+}$; (2) λ-*compatible* if $f \in R_{\tau}(\Gamma, x : \sigma)$ implies $\lambda f \in R_{\sigma \to \tau}^{\mathrm{val}}(\Gamma)$; (3) 0-*compatible* if $h \in R_{0}(\Gamma)$ implies $T!_{s[\![\sigma]\!]} \circ h \in R_{\tau}(\Gamma)$; and (4) +-*compatible* if the composite below is in $R_{\tau}(\Gamma, p : \sigma_1 + \sigma_2)$ whenever $h_i \in R_{\tau}(\Gamma, x_i : \sigma_i)$ for $i = 1, 2$ (the isomorphism is the canonical one from distributivity):

$$s[\![\Gamma, p : \sigma_1 + \sigma_2]\!] = s[\![\Gamma]\!] \times (s[\![\sigma_1]\!] + s[\![\sigma_2]\!]) \xrightarrow{\cong} \textstyle\sum_{i=1,2}(s[\![\Gamma]\!] \times s[\![\sigma_i]\!]) \xrightarrow{[h_1, h_2]} Ts[\![\tau]\!]$$

If R is λ-compatible, +-compatible and 0-compatible, we say R is $(\lambda, +, 0)$-*compatible.*

A hungry logical relation is one which 'eats' every morphism which preserves it: R is hungry if and only if $\big(f$ satisfies $R \implies f \in R_{\sigma}(\Gamma)\big)$ for every $f : s[\![\Gamma]\!] \to Ts[\![\sigma]\!]$ in $\mathcal{M}$.

6 CONCRETENESS

In Sec. 1.1.1 we obtained a well-pointed model by introducing a criterion which ensures the function spaces only consist of elements that are 'named' by a global element, namely *concreteness.* In this section we define concreteness in fibrational terms and show that a $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-model structure on $\mathrm{Krip}(p, \mathrm{N})$ induces a $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-model structure on the subcategory of concrete objects, for a fixed bi-CCC $\mathbb{C}$, fibration for logical relations $p : \mathbb{E} \to \mathbb{B}$, and product-preserving functor $\mathrm{N} : \mathbb{C} \to \mathbb{B}$. We shall then substantiate the claim in Sec. 1.1.1 that the function space in the subcategory of concrete objects consists only of those functions which preserve the relevant relations (Lemma 6.2).

Definition 6.1 (*cf.* Def. 1.1)**.** An object $(\underline{X}, R) \in \mathrm{Krip}(p, \mathrm{N})$ is *concrete* if every global element $g : 1 \to \underline{X}$ in $\mathbb{C}$ lifts to a global element $(g, \hat{g}) : (1, \top) \to (\underline{X}, R)$ lying over g. We write

$j : \mathrm{Conc}(p, \mathrm{N}) \hookrightarrow \mathrm{Krip}(p, \mathrm{N})$ for the full subcategory of concrete objects. When $p := \mathrm{cod}$ and $\mathrm{N} := \mathrm{N}_F$, so that $\mathrm{Krip}(p, \mathrm{N}) = \mathrm{Krip}_{\mathcal{M},F}$, we write $\mathrm{Conc}_{\mathcal{M},F}$ for $\mathrm{Conc}(p, \mathrm{N})$.

Notation 6.1. We reserve U for forgetful functors into the base category (either $\mathbb{C}$ or $\mathcal{M}$) and indicate the domain by a superscript, so that *e.g.* $\mathrm{U}^{\mathrm{Conc}(p,\mathrm{N})} : \mathrm{Conc}(p, \mathrm{N}) \to \mathbb{C}$.

Example 6.1 (*cf.* [O'Hearn and Riecke 1995]). Let $F : \mathbb{A} \to \mathbf{Fin}$. A Kripke relation $(\underline{X}, R) \in \mathrm{Krip}_{\mathbf{Fin},F}$ (Def. 2.2) is concrete if and only if $\Delta(\Gamma) := \{\lambda\gamma \in F\Gamma \,.\, x \,\big|\, x \in \underline{X}\} \subseteq R(\Gamma)$ for every $\Gamma \in \mathbb{A}$.

Every Kripke relation has a *concrete completion*: the inclusion $j : \mathrm{Conc}_{\mathcal{M},F} \hookrightarrow \mathrm{Krip}_{\mathcal{M},F}$ has a left adjoint defined by $\mathrm{K}(\underline{X}, R) = (\underline{X}, \mathrm{K}R)$, where $(\mathrm{K}R)(\Gamma) := R(\Gamma) \cup \Delta(\Gamma)$ (*cf.* also (2)). This is an instance of a general construction.

Definition 6.2. Let $(\underline{X}, R) \in \mathrm{Krip}(p, \mathrm{N})$. The set $\mathrm{CE}(\underline{X}, R)$ of *concrete extensions* consists of those $S \in \mathbb{E}$ such that $p(S) = \underline{X}$, $(\underline{X}, R) \le (\underline{X}, S)$ and $(\underline{X}, S)$ is concrete. The *concrete completion* $(\underline{X}, \widetilde{R})$ of $(\underline{X}, R)$ is the fibred product of all the concrete extensions: $(\underline{X}, \widetilde{R}) := \bigwedge_{S \in \mathrm{CE}(\underline{X},R)} (\underline{X}, S)$.

The concrete completion operation extends to a functor K which is left adjoint to j, thereby exhibiting $\mathrm{Conc}(p, \mathrm{N})$ as a reflective subcategory of $\mathrm{Krip}(p, \mathrm{N})$. For $\mathrm{Conc}(p, \mathrm{N})$ to inherit a monad from $\mathrm{Krip}(p, \mathrm{N})$, we further ask for $\mathrm{Conc}(p, \mathrm{N})$ to be a *coreflective* subcategory. This amounts to adding a *restriction* operation, taking an object of $\mathrm{Krip}(p, \mathrm{N})$ to the largest concrete subobject.

Definition 6.3. $\mathrm{Krip}(p, \mathrm{N})$ *admits a hull functor* if the inclusion $j : \mathrm{Conc}(p, \mathrm{N}) \hookrightarrow \mathrm{Krip}(p, \mathrm{N})$ has a right adjoint H. We denote the unit and counit of the adjunction $j \dashv \mathrm{H}$ by e and c, respectively, and write ϖ for the canonical map witnessing that H preserves products.

Example 6.2. In Ex. 6.1, the hull functor and its counit are given by the inclusion $\mathrm{H}\underline{X} \hookrightarrow \underline{X}$:

$$\mathrm{H}\underline{X} := \big\{x \in \underline{X} \,\big|\, \lambda\gamma \,.\, x \in R(\Gamma) \text{ for every } \Gamma \in \mathbb{A}\big\}, \qquad h \in (\mathrm{H}R)\Gamma \iff (F\Gamma \xrightarrow{h} \mathrm{H}\underline{X} \hookrightarrow \underline{X}) \in R(\Gamma).$$

In the preceding example, and in those we consider in Sec. 11, the action of the hull functor on relations is determined by the counit c. We axiomatise this with the next definition.

Definition 6.4. A hull functor H is *tractable* if the diagram below is a cartesian lifting for all $(\underline{X}, R)$ in $\mathrm{Krip}(p, \mathrm{N})$. We denote H's action by $(\underline{X}, R) \mapsto (\mathrm{H}\underline{X}, \mathrm{H}R)$ and the counit's components by $(\mathrm{c}, \hat{\mathrm{c}})$.

$$\begin{tikzcd}
j\mathrm{H}R \arrow[r, "\hat{\mathrm{c}}_{(\underline{X},R)}"] & R & \mathbb{E} \arrow[d, "p"] \\
\mathrm{N}j\mathrm{H}\underline{X} \arrow[r, "\mathrm{Nc}_{(\underline{X},R)}"'] & \mathrm{N}\underline{X} & \mathbb{B}
\end{tikzcd}$$

Example 6.3. Let $F : \mathbb{A} \to \mathcal{M}$. The hull $\mathrm{H} : \mathrm{Krip}_{\mathcal{M},F} \to \mathrm{Conc}_{\mathcal{M},F}$ is tractable iff $h \in (\mathrm{H}R)(\Gamma) \iff \mathrm{c}_X \circ h \in R(\Gamma)$ for all $(\underline{X}, R) \in \mathrm{Krip}_{\mathcal{M},F}$. Hull functors over **Fin** are always tractable (recall Ex. 6.2).

Remark 6.1. In the examples considered in this paper the (tractable) hull functor is always defined similarly to Ex. 6.2. This is because our models $\mathcal{M}$ are categories of sets-with-structure such that, if $X \in \mathcal{M}$ has carrier set $|X| \in \mathbf{Set}$, and $S \subseteq |X|$, then there exists a mono $\tilde{S} \rightarrowtail X$ in $\mathcal{M}$ such that $\tilde{S}$ has carrier S. In future work we shall seek a more widely-applicable condition (see Sec. 1.4).

When $\mathrm{Krip}(p, \mathrm{N})$ admits a hull functor the subcategory $\mathrm{Conc}(p, \mathrm{N})$ becomes a bi-CCC by the theory of (co)reflective subcategories (*e.g.* Adamek et al. [2009]; Borceux [1994]). Products and coproducts are inherited from $\mathrm{Krip}(p, \mathrm{N})$, and exponentials are given as follows:

$$(X \Rightarrow_{\mathrm{Conc}(p,\mathrm{N})} Y) := \mathrm{H}(jX \Rightarrow_{\mathrm{Krip}(p,\mathrm{N})} jY), \quad \mathrm{eval}^{\mathrm{Conc}(p,\mathrm{N})} := \mathrm{eval}^{\mathrm{Krip}(p,\mathrm{N})} \circ (\mathrm{c}_{jX \Rightarrow jY} \times jX)$$

Moreover, a strong monad $(\hat{T}, \hat{\mu}, \hat{\eta}, \hat{\mathrm{st}})$ on $\mathrm{Krip}(p, \mathrm{N})$ defines a monad $(\mathrm{W}, \mu^{\mathrm{W}}, \eta^{\mathrm{W}}, \mathrm{st}^{\mathrm{W}})$ on $\mathrm{Conc}(p, \mathrm{N})$:

$$\mathrm{W} := \mathrm{H}\hat{T}j, \qquad \mu^{\mathrm{W}} := \mathrm{H}\hat{T}\hat{\mu}j \circ \mathrm{H}\hat{T}\mathrm{c}\hat{T}j, \qquad \eta^{\mathrm{W}} := \mathrm{H}\hat{\eta}j \circ \mathrm{e}, \qquad \mathrm{st}^{\mathrm{W}} := \mathrm{H}\hat{\mathrm{st}} \circ \varpi \circ (\mathrm{e} \times \mathrm{H}\hat{T}j).$$

Lemma 6.1. *If* $\mathrm{Krip}(p, \mathrm{N})$ *admits a hull functor, then* (j, c) *is a morphism of strong monads* $\mathrm{W} \Rightarrow \hat{T}$*:*

$$\mathrm{c}_{\hat{T}j} \circ \mu^{\mathrm{W}} = \mu^{\hat{T}} \circ \hat{T}\mathrm{c}_{\hat{T}j} \circ \mathrm{c}_{\hat{T}j\mathrm{W}}, \qquad \mathrm{c}_{\hat{T}j} \circ \eta^{\mathrm{W}} = \eta^{\hat{T}}, \qquad \mathrm{c}_{\hat{T}j} \circ \mathrm{st}^{\mathrm{W}} = \mathrm{st}^{\hat{T}} \circ (\mathrm{id} \times \mathrm{c}_{\hat{T}j}).$$

Hence, if c *is component-wise monic, the monadic structure of* W *is determined by that of* $\hat{T}$*.*

The next result summarises this section.

Proposition 6.1. *For any bi-CCC* $\mathbb{C}$*, fibration for logical relations* $p : \mathbb{E} \to \mathbb{B}$ *and product-preserving functor* $\mathrm{N} : \mathbb{C} \to \mathbb{B}$ *such that* $\mathrm{Krip}(p, \mathrm{N})$ *admits a hull functor,* $\mathrm{Conc}(p, \mathrm{N})$ *is a bi-CCC and* j *strictly preserves products and coproducts. Moreover, if* $\hat{T}$ *is a strong monad on* $\mathrm{Krip}(p, \mathrm{N})$ *then* $\mathrm{H}\hat{T}j$ *is the underlying functor of a strong monad* W *on* $\mathrm{Conc}(p, \mathrm{N})$*, yielding the diagram below:*

$$\begin{tikzcd}
\mathrm{Conc}(p, \mathrm{N}) \arrow[loop left, "\mathrm{W} := \mathrm{H}\hat{T}j"] \arrow[r, ""] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] & \mathrm{Krip}(p, \mathrm{N}) \arrow[l, bend right, "\mathrm{K}"'] \arrow[l, bend left, "\mathrm{H}"] \arrow[loop right, "\hat{T}"]
\end{tikzcd} \qquad (5)$$

We emphasise that the inclusion j does *not* preserve exponentials. Instead, the exponential in $\mathrm{Conc}(p, \mathrm{N})$ is related to that in $\mathrm{Krip}(p, \mathrm{N})$ by the counit c. In the presence of slightly more structure—*e.g.* when p is the subobject fibration—the global elements of the carrier of the exponential in $\mathrm{Conc}(p, \mathrm{N})$ may be identified with morphisms in $\mathcal{M}$ preserving the relevant Kripke relations.

Lemma 6.2. *Suppose that, in the situation of Prop. 6.1, it is moreover the case that for every* $A \in \mathbb{B}$ *the (partially-ordered) fibre* $\mathbb{E}_A$ *over* A *has a bottom element* M_A*. Then* $\mathrm{U}^{\mathrm{Krip}(p,\mathrm{N})}$ *has a left adjoint* $L : \mathbb{C} \to \mathrm{Krip}(p, \mathrm{N}) : A \mapsto (A, M_A)$*. The composite* $\mathrm{K} \circ L$ *preserves the terminal object, and hence determines a natural isomorphism* $\mathcal{M}\big(1, \mathrm{U}^{\mathrm{Conc}(p,\mathrm{N})}(X \Rightarrow_{\mathrm{Conc}(p,\mathrm{N})} Y)\big) \cong \mathrm{Krip}(p, \mathrm{N})(jX, jY)$*.*

7 ABSTRACT OHR CONSTRUCTIONS FOR λ_{c}

We can now execute the strategy outlined in Sec. 2. We start with the *abstract* construction, in which we assume the properties we need on the indexing set; in Sec. 8 we show how to choose this set concretely. Fix a $\lambda_{\mathrm{c}}(\mathrm{S})$-model $(\mathcal{M}, T, s)$ and a small set $\mathbb{I}$ such that for each $i \in \mathbb{I}$ one has:

$$\text{A small category } \mathbb{A}_i\text{, a functor } F_i : \mathbb{A}_i \to \mathcal{M}\text{, and a monad lifting } \hat{T}_i \text{ of } T \text{ to } \mathrm{Krip}_{\mathcal{M},F_i}. \qquad (6)$$

For each $i \in \mathbb{I}$ one has $\mathrm{Krip}_{\mathcal{M},F_i} := \mathrm{Krip}(\mathrm{cod}, \mathrm{N}_{F_i})$ and its subcategory of concrete objects $\mathrm{Conc}_{\mathcal{M},F_i} := \mathrm{Conc}(\mathrm{cod}, \mathrm{N}_{F_i})$. Similarly, taking the fibration for logical relations $\prod_{i \in \mathbb{I}} \mathrm{cod}$ one obtains a category $\mathcal{K} := \mathrm{Krip}\big(\prod_{i \in \mathbb{I}} \mathrm{cod}, \langle \mathrm{N}_{F_i} \rangle_{i \in \mathbb{I}}\big)$ with objects denoted $X := (\underline{X}, \overline{X})$ as in Sec. 2, and a subcategory $\mathcal{C} := \mathrm{Conc}\big(\prod_{i \in \mathbb{I}} \mathrm{cod}, \langle \mathrm{N}_{F_i} \rangle_{i \in \mathbb{I}}\big)$. The structure in $\mathcal{K}$ and $\mathcal{C}$ is determined component-wise.

Lemma 7.1. *(1) Take* $(\underline{Y}, \overline{Y}) \in \mathcal{K}$*,* $f : \underline{X} \to \underline{Y}$ *in* $\mathcal{M}$*, and cartesian liftings in* $\mathcal{K}$ *and* $\mathrm{Krip}_{\mathcal{M},F_i}$*:*

$$\begin{tikzcd}
\prod_{i \in \mathbb{I}} \mathrm{Sub}(\widehat{\mathbb{A}_i}) \arrow[d, "\prod_i \mathrm{cod}"'] & f^*(\overline{Y}) \arrow[r, dashed] & \overline{Y} & f^*\big(\overline{Y}(i)\big) \arrow[r, dashed] & \overline{Y}(i) & \mathrm{Sub}(\widehat{\mathbb{A}_i}) \arrow[d, "\mathrm{cod}"] \\
\prod_{i \in \mathbb{I}} \widehat{\mathbb{A}_i} & \langle \mathrm{N}_{F_i} \rangle_i(\underline{X}) \arrow[r, "\langle \mathrm{N}_{F_i} \rangle_i(f)"'] & \langle \mathrm{N}_{F_i} \rangle_i(\underline{Y}) & \mathrm{N}_{F_i}(\underline{X}) \arrow[r, "\mathrm{N}_{F_i}(f)"'] & \mathrm{N}_{F_i}(\underline{Y}) & \widehat{\mathbb{A}_i}
\end{tikzcd}$$

Then $f^*(\overline{Y})$ *is determined component-wise, in the sense that* $\big(f^*(\overline{Y})\big)(i) = f^*\big(\overline{Y}(i)\big)$ *for all* $i \in \mathbb{I}$*.*

(2) Setting $\hat{T}(\underline{X}, \overline{X}) := \big(T\underline{X}, \overline{\hat{T}X}\big)$*, where* $\overline{\hat{T}X}(i) = \hat{T}_i\big(\overline{X}(i)\big)$ *for each* $i \in \mathbb{I}$*, defines a lifting of* T *to* $\mathcal{K}$*.*

(3) $(\underline{X}, \overline{X}) \in \mathcal{C}$ *if and only if* $\big(\underline{X}, \overline{X}(i)\big) \in \mathrm{Conc}_{\mathcal{M},F_i}$ *for every* $i \in \mathbb{I}$*.*

(4) $f : X \to Y$ *in* $\mathcal{C}$ *if and only if* $f : \big(\underline{X}, \overline{X}(i)\big) \to \big(\underline{Y}, \overline{Y}(i)\big)$ *in* $\mathrm{Conc}_{\mathcal{M},F_i}$ *for every* $i \in \mathbb{I}$*.*

For our abstract construction we want to assume enough structure so that Prop. 6.1 holds. This is captured by the following; assumption (3) axiomatises the situation of Ex. 6.2.

Assumption 7.1. We assume the following: (1) an interpretation $\hat{s}$ of base types, operations and primitives with $\mathrm{U}^{\mathcal{C}} \circ \hat{s} = s$; (2) an index $i_0 \in \mathbb{I}$ with $\mathbb{A}_{i_0} = \mathrm{Con}_{\mathrm{S}}$ and $F_{i_0} := \mathrm{U}^{\mathcal{C}} \circ \hat{s}[\![-]\!]$. Moreover, we assume: (3) $\mathcal{K}$ admits a tractable hull functor H with counit $\mathrm{c} : j\mathrm{H} \Rightarrow \mathrm{id}$ component-wise monic.

With these assumptions, Prop. 6.1 entails that $\mathcal{C}$ acquires a bi-CCC structure and a strong monad W with underlying functor $\mathrm{H}\hat{T}j$. We therefore recover the situation in Figure 1b.

Definition 7.1. We call $(\mathcal{C}, \mathrm{W}, \hat{s})$ the *abstract OHR model* on $(\mathcal{M}, T, s)$.

7.1 $\lambda_{\mathrm{c}}(\mathrm{S})$-Logical Relations Over $\mathcal{C}$

Having constructed $(\mathcal{C}, \mathrm{W}, \hat{s})$ we turn to considering logical relations over this model and relating them to logical relations over $(\mathcal{M}, T, s)$. To this end we consider the following two diagrams, in which $\hat{\mathrm{W}}$ is any lifting of W; recall that $\mathrm{Krip}(\mathrm{cod}, \mathrm{N}_{\mathrm{U}\circ\hat{s}[\![-]\!]})$ is exactly $\mathrm{Krip}_{\mathcal{M},F_{i_0}}$ by Assump. 7.1(2).

$$\begin{tikzcd}
\mathrm{Krip}_{\mathcal{C},\hat{s}[\![-]\!]} \arrow[loop left, "\hat{\mathrm{W}}"] \arrow[r] \arrow[d, "\mathrm{U}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Sub}(\widehat{\mathrm{Con}_{\mathrm{S}}}) \arrow[d, "\mathrm{cod}"] & & \mathrm{Krip}_{\mathcal{M},F_{i_0}} \arrow[loop left, "\hat{T}_{i_0}"] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Sub}(\widehat{\mathrm{Con}_{\mathrm{S}}}) \arrow[d, "\mathrm{cod}"] \\
\mathcal{C} \arrow[loop left, "\mathrm{W}"] \arrow[r, "\mathrm{N}_{\hat{s}[\![-]\!]}"'] & \widehat{\mathrm{Con}_{\mathrm{S}}} & & \mathcal{M} \arrow[loop left, "T"] \arrow[r, "\mathrm{N}_{\mathrm{U}\circ\hat{s}[\![-]\!]}"'] & \widehat{\mathrm{Con}_{\mathrm{S}}}
\end{tikzcd} \tag{7}$$

As observed in Sec. 2, for any $(\underline{X}, \mathcal{R}) \in \mathrm{Krip}_{\mathcal{C},\hat{s}[\![-]\!]}$ the faithfulness of U in (7) yields a chain of inclusions $\mathcal{R} \hookrightarrow \mathcal{C}(\hat{s}[\![-]\!], X) \hookrightarrow \mathcal{M}(\mathrm{U}\hat{s}[\![-]\!], \underline{X})$, so $(\underline{X}, \mathcal{R})$ becomes an object in $\mathrm{Krip}_{\mathcal{M},F_{i_0}}$.

Remark 7.1. The bi-CCC structure of $\mathrm{Krip}_{\mathcal{C},\hat{s}[\![-]\!]}$ is given as in Sec. 4.1, except one must take care to use the bi-CCC structure of $\mathcal{C}$. For example, the exponential $(X, \mathcal{R}) \Rightarrow_{\mathrm{Krip}_{\mathcal{C},\hat{s}[\![-]\!]}} (Y, \mathcal{S})$ is $\big(\underline{\mathrm{H}(X \Rightarrow Y)}, \overline{\mathrm{H}(X \supset Y)}, \mathcal{R} \ni \mathcal{S}\big)$, where $h \in (\mathcal{R} \ni \mathcal{S})\Gamma \iff \mathrm{eval} \circ \big\langle \mathrm{c}_{jX \Rightarrow jY} \circ h \circ \hat{s}[\![\rho]\!], u \big\rangle \in \mathcal{S}(\Delta)$ for any $\rho : \Gamma \to \Delta$ and $u \in \mathcal{R}(\Gamma)$; equivalently, $\mathrm{c}_{jX \Rightarrow jY} \circ h \in (\mathcal{R} \supset \mathcal{S})(\Gamma)$.

Together with the fact the forgetful functors $\mathrm{Krip}_{\mathcal{C},\hat{s}[\![-]\!]} \to \mathcal{C}$ and $\mathcal{C} \to \mathcal{M}$ both strictly preserve products, Remark 7.1 entails that—so long as $\hat{\mathrm{W}}$ interacts well with $\hat{T}_{i_0}$—then $\lambda_{\mathrm{c}}(\mathrm{S})$-logical relations over $\hat{\mathrm{W}}$ are 'tracked' by the cartesian closed structure of $\mathrm{Krip}_{\mathcal{M},F_{i_0}}$. The proof is by induction.

Lemma 7.2. *Let $\mathcal{R} = \{\mathcal{R}_\sigma\}_{\sigma \in \mathbf{Ty}}$ be a $\lambda_{\mathrm{c}}(\mathrm{S})$-logical relation over $\hat{\mathrm{W}}$ such that: (1) $h \in \mathcal{R}_\sigma(\Gamma)$ if and only if $\mathrm{c}_{\hat{T} j\hat{s}[\![\sigma]\!]} \circ h \in (\hat{T}_{i_0} \mathcal{R}^{\mathrm{val}}_\sigma)(\Gamma)$; and (2) $\overline{\hat{s}[\![\beta]\!]}(i_0) = \mathcal{R}^{\mathrm{val}}_\beta$ for every $\beta \in \mathbf{Ty}$. Then $\overline{\hat{s}[\![\sigma]\!]}(i_0) = \mathcal{R}^{\mathrm{val}}_\sigma$ and $(\overline{\mathrm{W}\hat{s}[\![\sigma]\!]})(i_0) = \mathcal{R}_\sigma$ for every type $\sigma \in \mathbf{Ty}$.*

We now want to find $\hat{\mathrm{W}}$ and $\hat{T}_{i_0}$ so that condition (1) in this lemma holds automatically. Recalling Prop. 5.1, it is natural to use the $\top\top$-lifting arising from the given logical relation, namely $\hat{\mathrm{W}} = \mathrm{W}^{\top\top[\mathcal{R}]}$. It remains to identify a suitable choice of $\hat{T}_{i_0}$. To this end, note that every $(\mathrm{W}\hat{s}[\![\sigma]\!], \mathcal{R}_\sigma)$ gives rise to a $\top\top$-lifting parameter $(T\underline{\hat{s}[\![\sigma]\!]}, \langle\mathcal{R}_\sigma\rangle)$ in $\mathrm{Krip}_{\mathcal{M},F_{i_0}}$ as follows:

$$\langle\mathcal{R}_\sigma\rangle(\Gamma) := \big\{\mathrm{c}_{\hat{T} j\hat{s}[\![\sigma]\!]} \circ h \,\big|\, h \in \mathcal{R}_\sigma(\Gamma)\big\} \subseteq \mathcal{M}(\underline{\hat{s}[\![\Gamma]\!]}, T\underline{\hat{s}[\![\sigma]\!]})$$

Lemma 7.3. *Let $\mathcal{R} = \{\mathcal{R}_\sigma\}_{\sigma \in \mathbf{Ty}}$ be a hungry $\lambda_{\mathrm{c}}(\mathrm{S})$-logical relation over $\hat{\mathrm{W}} := \mathrm{W}^{\top\top[\mathcal{R}]}$. Then condition (1) of Lemma 7.2 holds: $h \in \mathcal{R}_\sigma(\Gamma) \iff \mathrm{c}_{\hat{T} j\hat{s}[\![\sigma]\!]} \circ h \in (T^{\top\top[\langle\mathcal{R}\rangle]} \mathcal{R}^{\mathrm{val}}_\sigma)(\Gamma)$.*

Combining Lemmas 7.2 and 7.3 yields the following, which makes precise the sense in which morphisms in the abstract OHR model preserve every compatible logical relation over $\mathcal{C}$.

Proposition 7.1. *Let $\mathcal{R} = \{\mathcal{R}_\sigma\}_{\sigma \in \mathbf{Ty}}$ be a hungry $\lambda_{\mathrm{c}}(\mathrm{S})$-logical relation over $\hat{\mathrm{W}} := \mathrm{W}^{\top\top[\mathcal{R}]}$ and suppose $i_0 \in \mathbb{I}$ is such that $\hat{T}_{i_0} = T^{\top\top[\langle\mathcal{R}\rangle]}$ and $\overline{\hat{s}[\![\sigma]\!]}(i_0)(\beta) = \mathcal{R}^{\mathrm{val}}_\beta$ for every $\beta \in \mathbf{B}$. Then $\overline{\hat{s}[\![\sigma]\!]}(i_0) = \mathcal{R}^{\mathrm{val}}_\sigma$ and $(\overline{\mathrm{W}\hat{s}[\![\sigma]\!]})(i_0) = \mathcal{R}_\sigma$ for every type $\sigma \in \mathbf{Ty}$, so every $f : \hat{s}[\![\Gamma]\!] \to \mathrm{W}\hat{s}[\![\sigma]\!]$ in $\mathcal{C}$ satisfies $\mathcal{R}$.*

8 A FULLY ABSTRACT MODEL FOR λ_c

Choosing the Indexing Set $\mathbb{I}$. We now follow the strategy sketched in Sec. 2.2. To instantiate the abstract OHR construction (Def. 7.1) we need to choose $\mathbb{I}$, $i_0 \in \mathbb{I}$, and $\hat{s}$ so that Assump. 7.1 and the hypotheses of Prop. 7.1 hold. Thus, we replace Assump. 7.1 with the weaker assumptions below.

Assumption 8.1. Let $(\mathcal{M}, T, s)$ be a $\lambda_c^+(\mathbf{S})$-model such that $\mathcal{M}$ is *small* and, for any small set $\mathbb{J}$ and $\mathbb{J}$-indexed set of functors $\mathbb{A}_j \to \mathcal{M}$, the category $\mathrm{Krip}(\prod_{j\in\mathbb{J}} \mathrm{cod}, \langle \mathrm{N}_{F_j} \rangle_{j\in\mathbb{J}})$ admits a tractable hull functor with counit c component-wise monic.

Remark 8.1. The size restriction is not onerous. Although many models of interest are large (*e.g.* **Set**, **ωCpo**, presheaf categories), one generally works within a small subcategory. For example, one may replace **Set** with the subcategory $\mathbf{Set}_\kappa$ of *hereditarily-κ sets*, for κ some infinite cardinal.

We construct $\mathbb{I}$ in stages, ranging over the data required to construct a λ_c^+-model structure on each category of Kripke relations $\mathcal{K}_i$. By Prop. 5.1 this amounts to ranging over a collection of $\lambda_c(\mathbf{S})$-logical relations. First we give the data necessary to construct $\mathcal{K}_i$:

- Fix a set $\mathbb{S}$ of small categories such that $\mathrm{Con}_{\mathbf{S}}^{\mathrm{op}} \in \mathbb{S}$ (the singleton $\{\mathrm{Con}_{\mathbf{S}}^{\mathrm{op}}\}$ suffices).
- For each $\mathbb{A} \in \mathbb{S}$, let $\mathrm{Fun}(\mathbb{A}) := [\mathbb{A}, \mathcal{M}]$ be the set of functors $\mathbb{A} \to \mathcal{M}$; this is small since $\mathcal{M}$ is.

Since the nerve functor $\mathrm{N}_F : \mathcal{M} \to \widehat{\mathbb{A}}$ preserves limits for any $F : \mathbb{A} \to \mathcal{M}$, for each $\mathbb{A} \in \mathbb{S}$ and $F \in \mathrm{Fun}(\mathbb{A})$ one obtains a bi-CCC $\mathrm{Krip}_{\mathcal{M},F}$. We then take

- For each $\mathbb{A} \in \mathbb{S}$ and $F \in \mathrm{Fun}(\mathbb{A})$, take $\mathrm{Lift}(\mathbb{A}, F)$ to be the set of liftings of T to $\mathrm{Krip}_{\mathcal{M},F}$. This is small because $\mathcal{M}$ is small and $\mathrm{Krip}_{\mathcal{M},F} \to \mathcal{M}$ has small fibres.

Clearly $\mathrm{U} \circ \hat{s}\llbracket - \rrbracket \in \mathrm{Fun}(\mathrm{Con}_{\mathbf{S}}^{\mathrm{op}})$, so Assump. 7.1(2) will hold, and $T^{\top\top[\langle\textsc{def}\rangle]} \in \mathrm{Lift}(\mathrm{Con}_{\mathbf{S}}^{\mathrm{op}}, \mathrm{U}\hat{s}\llbracket - \rrbracket)$.

It remains to construct the interpretation $\hat{s}$. By Lemma 7.1 it suffices to work component-wise, so fix $\mathbb{A} \in \mathbb{S}$, $F \in \mathrm{Fun}(\mathbb{A})$ and $\hat{T} \in \mathrm{Lift}(\mathbb{A}, F)$. We need each base type to be interpreted by a concrete object, so we range over interpretations of base types $r : \mathbf{B} \to \mathrm{Conc}_{\mathcal{M},F}$ lying over the interpretation s. However, to interpret primitives with thunks we want to use the exponentials of $\mathrm{Krip}_{\mathcal{M},F}$, which lie over those in $\mathcal{M}$, rather than those of $\mathrm{Conc}_{\mathcal{M},F}$. Thus, we use the composite $j \circ r : \mathbf{B} \to \mathrm{Krip}_{\mathcal{M},F}$. This extends canonically to an interpretation $(j \circ r)\llbracket - \rrbracket$ of all types, so we take

- $\mathrm{Interp}(\mathbb{A}, F, \hat{T})$ is the set of all maps $r : \mathbf{B} \to \mathrm{Conc}_{\mathcal{M},F}$ such that: (1) $\mathrm{U}^{\mathrm{Conc}_{\mathcal{M},F}} \circ r = s$; and (2) $s\llbracket \mathrm{op} \rrbracket : jr\llbracket \alpha \rrbracket \to \hat{T}(jr\llbracket \kappa \rrbracket)$ and $s\llbracket \xi \rrbracket : 1 \to (jr)\llbracket \kappa \rrbracket$ for each $\mathrm{op} : \alpha \rightsquigarrow \kappa$ and $\xi : \kappa$ in $\mathbf{S}$.

This set is small because the fibration $\mathrm{Krip}_{\mathcal{M},F} \to \mathcal{M}$ has small fibres, so for each $\beta \in \mathbf{B}$ there's a small set of choices of object lying over $s\llbracket \beta \rrbracket$. Putting everything together, we define

$$\mathbb{I} := \left\{ (\mathbb{A}, F, \hat{T}, r) \,\middle|\, \mathbb{A} \in \mathbb{S}, F \in \mathrm{Fun}(\mathbb{A}), \hat{T} \in \mathrm{Lift}(\mathbb{A}, F), r \in \mathrm{Interp}(\mathbb{A}, F, \hat{T}) \right\} \tag{8}$$

We now detail the OHR construction. Following O'Hearn and Riecke [1995], set $\hat{s}(\beta) = \left(s(\beta), \overline{\hat{s}(\beta)}\right)$ where $\overline{\hat{s}(\beta)}(\mathbb{A}, F, \hat{T}, r)$ is the second projection of $r(\beta)$; this is a mapping $\mathbf{B} \to \mathcal{C}$ by Lemma 7.1(3). To interpret operations and primitives we use the following lemma, which shows that these lift from $\mathcal{M}$ to $\mathcal{K}$; we can then use the adjunction $j \dashv \mathrm{H}$ to construct the required maps in $\mathcal{C}$. Thunks require particular care because if $\sigma_i = (1 \to \gamma_i)$ then $\hat{s}\llbracket \sigma_i \rrbracket = \mathrm{H}(1 \Rightarrow j\mathrm{W}\hat{s}\llbracket \gamma_i \rrbracket)$. We therefore define an object $\ulcorner \hat{s}\llbracket \sigma_i \rrbracket \urcorner$ and map $m_{\sigma_i} : j\hat{s}\llbracket \sigma_i \rrbracket \to \ulcorner \hat{s}\llbracket \sigma_i \rrbracket \urcorner$ by setting $\ulcorner \hat{s}\llbracket \sigma_i \rrbracket \urcorner := \hat{s}\llbracket \gamma_i \rrbracket$ and $m_{\sigma_i} := \mathrm{id}$ if $\sigma_i = \gamma_i$, and $\ulcorner \hat{s}\llbracket \sigma_i \rrbracket \urcorner := 1 \Rightarrow \hat{T} j\hat{s}\llbracket \gamma_i \rrbracket$ and $m_{\sigma_i} := (\mathrm{id} \Rightarrow \mathrm{c}_{\hat{T} j\hat{s}\llbracket \gamma_i \rrbracket}) \circ \mathrm{c}_{1 \Rightarrow \mathrm{W}\hat{s}\llbracket \gamma_i \rrbracket}$ if σ_i is a thunk $(1 \to \gamma_i)$.

Lemma 8.1. *For $\mathbb{I}$ and $\hat{s} : \mathbf{B} \to \mathcal{C}$ as defined above, and $\mathcal{K}$ and $\mathcal{C}$ constructed as in Figure 1b: (1) if $\mathrm{op} : \alpha \rightsquigarrow \kappa$, then $s\llbracket \mathrm{op} \rrbracket : j\hat{s}\llbracket \alpha \rrbracket \to \hat{T}\hat{s}\llbracket \kappa \rrbracket$ in $\mathcal{K}$; and (2) if $\xi : \gamma$, then $s\llbracket \xi \rrbracket : 1 \to \hat{s}\llbracket \gamma \rrbracket$ in $\mathcal{C}$; and (3) if $\xi : \sigma_1 * \cdots * \sigma_n \to \gamma$, then $s\llbracket \xi \rrbracket : 1 \to \left(j\ulcorner \hat{s}\llbracket \sigma_1 \rrbracket \urcorner \times \cdots \times j\ulcorner \hat{s}\llbracket \sigma_n \rrbracket \urcorner\right) \Rightarrow \hat{T}\hat{s}\llbracket \gamma \rrbracket\big)$ in $\mathcal{K}$.*

For the interpretation of primitives note that for any category $\mathrm{Krip}(p, \mathrm{N})$ with hull functor H there is a bijective correspondence between maps $1 \longrightarrow \mathrm{H}(jX \Rightarrow j\mathrm{H}\hat{T}jY)$ and maps $1 \times jX \longrightarrow \hat{T}jY$.

Definition 8.1. The *OHR model* $(\textsc{ohr}(\mathcal{M}), \mathrm{W}, \hat{s})$ over $(\mathcal{M}, T, s)$ is the $\lambda_c(\mathbf{S})$-model obtained by instantiating Def. 7.1 with: (1) indexing set $\mathbb{I}$ as in (8), and $\mathbb{A}_i, F_i$ and $\hat{T}_i$ given by the projections; and (2) interpretation $\hat{s}$ with $\underline{\hat{s}(\beta)} := s(\beta)$ and $\overline{\hat{s}(\beta)}(\mathbb{A}, F, \hat{T}, r)$ the second projection of $r(\beta)$; and (3) operations $\mathrm{op} : \alpha \rightsquigarrow \kappa$ interpreted by $\hat{s}[\![\mathrm{op}]\!] := \mathrm{H}(s[\![\mathrm{op}]\!]) \circ \mathrm{e}_{\hat{s}[\![\alpha]\!]}$, primitives $(\xi : \gamma)$ by $\hat{s}[\![\xi]\!] := s[\![\xi]\!]$ and primitives $(\xi : \sigma_1 * \cdots * \sigma_n \longrightarrow \gamma)$ by setting $\hat{s}[\![\xi]\!]$ to be the arrow corresponding to $\mathrm{eval} \circ (s[\![\xi]\!] \times \prod_{i=1}^{n} m_{\sigma_i})$ across the bijective correspondence above.

The next result makes precise the idea that $\textsc{ohr}(\mathcal{M})$-morphisms preserve every compatible logical relation over $\textsc{ohr}(\mathcal{M})$. For the proof, set $i_0 := \big(\mathrm{Con}_{\mathrm{S}}^{\mathrm{op}}, \mathrm{U} \circ \hat{s}[\![-]\!], T^{\top\top[\langle\mathcal{R}\rangle]}, r\big)$ with $r : \beta \mapsto (s[\![\beta]\!], R_{\beta}^{\mathrm{val}})$, show that $i_0 \in \mathbb{I}$, *i.e.* that $r \in \mathrm{Interp}\big(\mathrm{Con}_{\mathrm{S}}^{\mathrm{op}}, \mathrm{U} \circ \hat{s}[\![-]\!], T^{\top\top[\langle\mathcal{R}\rangle]}\big)$, and apply Prop. 7.1.

Proposition 8.1. *Let $\mathcal{R} = \{\mathcal{R}_\sigma\}_{\sigma \in \mathrm{Ty}}$ be a hungry, λ-compatible $\lambda_c(\mathbf{S})$-logical relation over $\hat{\mathrm{W}} := \mathrm{W}^{\top\top[\mathcal{R}]}$ such that every global element $g : 1 \longrightarrow s[\![\beta]\!]$ in $\mathcal{M}$ is in $\mathcal{R}_{\beta}^{\mathrm{val}}(\diamond)$. Then any morphism $f : \hat{s}[\![\Gamma]\!] \longrightarrow \mathrm{W}\hat{s}[\![\sigma]\!]$ in $\textsc{ohr}(\mathcal{M})$ satisfies $\mathcal{R}$, and hence—since $\mathcal{R}$ is hungry—is in $\mathcal{R}_\sigma(\Gamma)$.*

Together with Prop. 5.1, the preceding entails $\textsc{ohr}(\mathcal{M})$ is "saturated": if there exist enough global elements, no $\lambda_c(\mathbf{S})$-morphism can cut out any morphisms. This should be contrasted with Ex. 1.2.

Corollary 8.1. *If $F : (\mathcal{N}, S, t) \longrightarrow (\textsc{ohr}(\mathcal{M}), \mathrm{W}, \hat{s})$ is any strict $\lambda_c(\mathbf{S})$-morphism such that the induced composite $(\mathrm{U}^{\mathrm{OHR}(\mathcal{M})} \circ F)_{1, t[\![\beta]\!]} : \mathcal{N}(1, t[\![\beta]\!]) \longrightarrow \mathcal{M}(1, s[\![\beta]\!])$ is onto, then every $f : \hat{s}[\![\Gamma]\!] \longrightarrow \mathrm{W}\hat{s}[\![\sigma]\!]$ in $\textsc{ohr}(\mathcal{M})$ is in the image of F.*

Full Completeness of the OHR Construction. Since $\textsc{def}$ is always $\lambda_c(\mathbf{S})$-logical over $T^{\top\top[\textsc{def}]}$ (Thm. 5.1), the final obstacle to full completeness is the concreteness condition on global elements. For this we relate the induced interpretation $\hat{s}$ to s. Write $\mathcal{D}_\sigma(\Gamma) := \big\{f \,\big|\, f \text{ is definable in } (\mathcal{M}, T, s)\big\}$ for any context Γ and $\sigma \in \mathbf{Ty}$, and call a context Γ *ground* if $\sigma \in \mathbf{G}$ whenever $(x : \sigma) \in \Gamma$. Since the forgetful functor $\mathrm{U}^{\mathrm{OHR}(\mathcal{M})}$ strictly preserves products, $\underline{\hat{s}[\![\Gamma]\!]} = s[\![\Gamma]\!]$ for any ground context Γ.

Lemma 8.2. *For every ground context Γ and ground type γ, $\mathrm{U}^{\mathrm{OHR}(\mathcal{M})}\big(\mathrm{c}_{\hat{T}j\hat{s}[\![\gamma]\!]} \circ \hat{s}[\![\Gamma \vdash M : \gamma]\!]\big) = s[\![\Gamma \vdash M : \gamma]\!]$. Hence $\textsc{def}$ and $\mathcal{D}$ coincide on closed terms of ground type: $\textsc{def}_{\gamma}^{\mathrm{val}}(\diamond) = \mathcal{D}_{\gamma}^{\mathrm{val}}(\diamond)$.*

Remark 8.2. It follows that our semantic definition of contextual equivalence is consistent between $(\mathcal{M}, T, s)$ and $(\textsc{ohr}(\mathcal{M}), \mathrm{W}, \hat{s})$: we have $M \simeq_{\mathrm{ctx}} M'$ in $\mathcal{M}$ if and only $M \simeq_{\mathrm{ctx}} M'$ in $\textsc{ohr}(\mathcal{M})$.

By Lemma 8.2, if a global element $g : 1 \longrightarrow s[\![\beta]\!]$ in $\mathcal{M}$ is such that $\eta_{s[\![\beta]\!]} \circ g$ is definable, then this composite is also definable in $\textsc{ohr}(\mathcal{M})$. Hence from Prop. 8.1 we obtain full completeness.

Theorem 8.1. *Let $(\mathcal{M}, T, s)$ be a small, well-pointed $\lambda_c(\mathbf{S})$-model such that for every $\beta \in \mathbf{B}$ and global element $g : 1 \longrightarrow s[\![\beta]\!]$ the composite $\eta_{s[\![\beta]\!]} \circ g$ is definable. Then, if $\mathrm{Krip}(\prod_i \mathrm{cod}, \langle F \rangle_{i \in \mathbb{I}})$ admits a tractable hull functor and the counit of the adjunction $j \dashv \mathrm{H}$ is component-wise monic, the induced model $(\textsc{ohr}(\mathcal{M}), \mathrm{W}, \hat{s})$ is well-pointed and fully complete, hence fully abstract.*

Example 8.1. The theorem applies to any model on **Set** in which the signature has a primitive $\underline{b} : \beta$ for every $b \in s(\beta)$ and $\beta \in \mathbf{B}$ (for the size restriction take $\mathbf{Set}_\kappa$ for a large enough κ, *cf.* Remark 8.1), such as when one has a base type nat and primitives $\underline{n}$: nat for every $n \in \mathbb{N}$.

9 EXAMPLE: A FULLY ABSTRACT MODEL FOR IMMUTABLE STATE

The structure of the OHR model is closely related to that of the original model, and one can use this to compute in the OHR model. We highlight these properties by returning to Ex. 1.1: let us denote the signature considered there by S_{RO} and write $(\mathbf{Fin}, \mathrm{R}, s)$ for the associated semantic model. We shall show why the counterexample κ, which shows that $(\mathbf{Fin}, \mathrm{R}, s)$ is not fully abstract, does not give rise to a counterexample in $(\textsc{ohr}(\mathbf{Fin}), \mathrm{W}, \hat{s})$; along the way we shall see a particular example of how Prop. 8.1 ensures full completeness of the OHR model.

We begin by making explicit the structure of the OHR model for a $\lambda_c(\mathbf{S})$-model $(\mathcal{M}, T, s)$ with $\mathcal{M} \subseteq \mathbf{Set}$. Recalling Ex. 6.2, the counit c is an inclusion on carrier sets: for any $X = (\underline{X}, \overline{X}) \in \mathcal{K}$ one has $\mathrm{c}_X : \underline{jHX} \hookrightarrow \underline{X}$. All the λ_c-model structure of $\textsc{ohr}(\mathcal{M})$ is then determined by c and the corresponding structure on $\mathcal{M}$. For example, products in $\textsc{ohr}(\mathcal{M})$ coincide with products in $\mathcal{M}$ and $\mathrm{eval}^{\textsc{ohr}(\mathcal{M})}$ is the following composite in $\mathcal{M}$, so that $\mathrm{eval}^{\textsc{ohr}(\mathcal{M})}(f, x) = f(x)$:

$$\underline{\mathrm{H}(jX \Rightarrow jY)} \times \underline{X} \hookrightarrow (\underline{X} \Rightarrow \underline{Y}) \times \underline{X} \xrightarrow{\mathrm{eval}} \underline{Y} \tag{9}$$

By Lemma 6.1, similar remarks apply to the monadic structure, *e.g.* $\eta^{\mathrm{W}}_X(x) = \eta^T_{\underline{X}}(x)$ for all $x \in \underline{X}$.

We now return to the particular case of Ex. 1.1. Since $s[\![\mathrm{tt}]\!]$ and $s[\![\mathrm{ff}]\!]$ name the two elements of $s[\![\mathrm{bool}]\!]$, the model $(\mathbf{Fin}, \mathrm{R}, s)$ satisfies the conditions of Thm. 8.1. Hence the OHR model $(\textsc{ohr}(\mathbf{Fin}), \mathrm{W}, \hat{s})$ on $(\mathbf{Fin}, \mathrm{R}, s)$ exists and is fully abstract.

Using the relationship between $\textsc{ohr}(\mathbf{Fin})$ and $\mathbf{Fin}$ sketched above, one sees the following.

Example 9.1. For any closed $\lambda_c(\mathrm{S}_{\mathrm{RO}})$-terms $N, N' : 1 \to \mathrm{bool}$ and variable $f : (1 \to \mathrm{bool}) \to \mathrm{bool}$,

$$\mathrm{c}_{\hat{T} j \hat{s}[\![\mathrm{bool}]\!]} \circ \hat{s}[\![f : (1 \to \mathrm{bool}) \to \mathrm{bool} \vdash \mathrm{or}(\langle f\,N, f\,N' \rangle) : \mathrm{bool}]\!] = \lambda\varphi\,.\,\lambda i\,.\,\varphi(n(i))(i) \vee \varphi(n'(i))(i)$$

where $n = \mathrm{c}_{\hat{T} j \hat{s}[\![1 \to \mathrm{bool}]\!]} \circ \hat{s}[\![N]\!]$ and $n' = \mathrm{c}_{\hat{T} j \hat{s}[\![1 \to \mathrm{bool}]\!]} \circ \hat{s}[\![N']\!]$. Since the components of c are injections, this determines $\hat{s}[\![\mathrm{or}(\langle f\,N, f\,N' \rangle)]\!]$. Similar considerations show that, where M and M' are as in (1), then the set maps $\hat{s}[\![M]\!]$ and $\hat{s}[\![M']\!]$ have the same action as $s[\![M]\!]$ and $s[\![M']\!]$.

We can now show why the counterexample morphism κ of Ex. 1.1 cannot determine an element of $\hat{s}[\![(1 \to \mathrm{bool}) \to \mathrm{bool}]\!]$. The idea, which holds quite generally, is to instantiate a version of Figure 1a with $\mathbf{Fin}$ replaced by $\textsc{ohr}(\mathbf{Fin})$, then adapt the argument from Ex. 1.2 to show κ does not restrict to a map in $\textsc{ohr}(\mathbf{Fin})$. Accordingly, let $\mathbb{L}_{\textsc{ohr}(\mathbf{Fin})}$ be the category obtained by change-of-base along $\Delta \circ \mathrm{U}^{\textsc{ohr}(\mathcal{M})} : \textsc{ohr}(\mathcal{M}) \to \mathbf{Fin} \times \mathbf{Fin}$. Explicitly, $\mathbb{L}_{\textsc{ohr}(\mathbf{Fin})}$ has objects triples $\big((\underline{X}, \overline{X}), R_1, R_2\big)$ where each R_i is a binary relation on the carrier $\underline{X}$; products and exponentials are defined as in $\mathbb{L}_{\mathbf{Fin}}$.

For the monad, the universal property of the fibration induces a lifting $\hat{\mathrm{W}}$ of W to $\mathbb{L}_{\textsc{ohr}(\mathbf{Fin})}$: for $\big((\underline{X}, \overline{X}), S\big) \in \mathbb{L}_{\textsc{ohr}(\mathbf{Fin})}$ one has $(\underline{X}, S) \in \mathbb{L}_{\mathbf{Fin}}$ and $(\mathrm{R}\underline{X}, \hat{\mathrm{R}}S) \in \mathbb{L}_{\mathbf{Fin}}$, so one defines $\hat{\mathrm{W}}S$ using the cartesian lifting in (10), below. Explicitly, $(h, h') \in (\hat{\mathrm{W}}R)_i \iff (h\,i, h'\,i) \in R_i$. To finish defining a the semantic model, set $\hat{t}(\mathrm{bool}) = \big(\hat{s}[\![\mathrm{bool}]\!], \{(0,0), (1,1)\}, \{(0,0), (1,1)\}\big)$ and interpret primitives and operations using the universal property of the cartesian lifting. Then $(\mathbb{L}_{\textsc{ohr}(\mathbf{Fin})}, \hat{\mathrm{W}}, \hat{t})$ is a $\lambda_c(\mathrm{S}_{\mathrm{RO}})$-model and the forgetful functor $\mathbb{L}_{\textsc{ohr}(\mathbf{Fin})} \to \textsc{ohr}(\mathbf{Fin})$ is a $\lambda_c(\mathrm{S}_{\mathrm{RO}})$-morphism.

$$\begin{tikzcd}
\hat{\mathrm{W}}S \arrow[r, dashed] & \hat{\mathrm{R}}S & & \mathrm{Sub}(\mathbf{Fin}) \times \mathrm{Sub}(\mathbf{Fin}) \arrow[d, "\mathrm{cod} \times \mathrm{cod}"] \\
\mathrm{N}\underline{\mathrm{W}X} \arrow[r, "\mathrm{N}\mathrm{c}_{\hat{T} jX}"'] & \mathrm{NR}(\underline{X}) & & \mathbf{Fin} \times \mathbf{Fin}
\end{tikzcd} \tag{10}$$

We can now show the counterexample κ from Ex. 1.1 does not restrict to a counterexample κ' in $\textsc{ohr}(\mathbf{Fin})$, *i.e.* that there does not exist κ' in $\textsc{ohr}(\mathbf{Fin})$ and $i \in 2$ such that

$$\kappa = \Big(1 \xrightarrow{\kappa'} \underline{\mathrm{W}\hat{s}[\![(1 \to \mathrm{bool}) \to \mathrm{bool}]\!]} \hookrightarrow \mathrm{R}\big((1 \Rightarrow \mathrm{R}2) \Rightarrow \mathrm{R}2\big) \xrightarrow{\mathrm{eval}_i} (1 \Rightarrow \mathrm{R}2) \Rightarrow \mathrm{R}2\Big).$$

If such a κ' existed, it must lift to $\mathbb{L}_{\textsc{ohr}(\mathbf{Fin})}$ by Cor. 8.1, but, arguing in the same way as Ex. 1.2, one sees this is impossible. Similar reasoning remedies our omission in Ex. 1.1.

Lemma 9.1. *For M, M' as in Ex. 1.1, $M \simeq_{\mathrm{ctx}} M'$ in $(\mathbf{Fin}, \mathrm{R}, s)$.*

Proof. By Remark 8.2 it suffices to show $M \simeq_{\mathrm{ctx}} M'$ in the OHR model. Suppose for a contradiction that $\hat{s}\llbracket M \rrbracket \neq \hat{s}\llbracket M' \rrbracket$. The counit determines an inclusion $\underline{\mathrm{W}\hat{s}\llbracket ((1 \to \mathrm{bool}) \to \mathrm{bool}) \to \mathrm{bool} \rrbracket} \hookrightarrow \mathrm{R}(\underline{\hat{s}\llbracket (1 \to \mathrm{bool}) \to \mathrm{bool} \rrbracket} \Rightarrow \mathrm{R}\mathbb{2})$ so there exists some $i \in \mathbb{2}$ and $\delta \in \underline{\hat{s}\llbracket (1 \to \mathrm{bool}) \to \mathrm{bool} \rrbracket}$ such that $\hat{s}\llbracket M \rrbracket(i)(\delta) \neq \hat{s}\llbracket M' \rrbracket(i)(\delta)$. Using the interpretations of read, tt, ff and ¬ in $\textsc{ohr}(\mathcal{M})$ one sees that $\underline{\hat{s}\llbracket (1 \to \mathrm{bool}) \to \mathrm{bool} \rrbracket}$ has four elements, and so must equal $s\llbracket (1 \to \mathrm{bool}) \to \mathrm{bool} \rrbracket$. A long but basic check then shows that if $s\llbracket M \rrbracket(i)(\omega) \neq s\llbracket M' \rrbracket(i)(\omega)$ then ω does not lift to a morphism in $\mathbb{L}_{\mathbf{Fin}}$ so, by Ex. 9.1, δ cannot lift to a morphism in $\mathbb{L}_{\mathbf{Fin}}$. On the other hand, by concreteness and Cor. 8.1, δ defines a morphism $1 \to \hat{t}\llbracket (1 \to \mathrm{bool}) \to \mathrm{bool} \rrbracket$ in $\mathbb{L}_{\textsc{ohr}(\mathbf{Fin})}$. But then $\delta : (X, R_1, R_2) \to (Y, S_1, S_2)$ in $\mathbb{L}_{\textsc{ohr}(\mathbf{Fin})}$ implies $\delta : (\underline{X}, R_1, R_2) \to (\underline{Y}, S_1, S_2)$ in $\mathbb{L}_{\mathbf{Fin}}$, contradicting the preceding. □

10 A FULLY ABSTRACT MODEL FOR λ_{c}^{+}

We now fold sum types into the development of Secs. 7 and 8. We saw in Sec. 5 that to characterise definability with sum types one needs to restrict to $\hat{T}$-closed objects. Accordingly, for our OHR construction we restrict not just to concrete objects, but to those that are both concrete and $\hat{T}$-closed. Write $k : \mathrm{ConcCl}(p, \mathrm{N}, \hat{T}) \hookrightarrow \mathrm{Krip}(p, \mathrm{N})$ for the subcategory of concrete, $\hat{T}$-closed objects. We induce structure on this category as we did for the concrete and $\hat{T}$-closed subcategories.

Lemma 10.1 (*cf.* Lemma 4.4 and Prop. 6.1)**.** *In the situation of Prop. 6.1:*

(1) The closure operator $\mathrm{id}^{\top\top[\hat{T}]}$ *restricts to* $\mathrm{Conc}(p, \mathrm{N})$*: if* (A, R) *is concrete, then so is* $\mathrm{id}^{\top\top[\hat{T}]}(A, R)$*. Hence* $\mathrm{ConcCl}(p, \mathrm{N}, \hat{T})$ *is a replete reflective subcategory of* $\mathrm{Conc}(p, \mathrm{N})$*.*
(2) $\mathrm{ConcCl}(p, \mathrm{N}, \hat{T})$ *is an* exponential ideal *(e.g. Johnstone [2002, p. 52]) of* $\mathrm{Conc}(p, \mathrm{N})$*.*
(3) If H *is tractable, then* $\mathrm{H}\hat{T}j$ *restricts to a strong monad on* $\mathrm{ConcCl}(p, \mathrm{N}, \hat{T})$*.*

Hence, $\mathrm{ConcCl}(p, \mathrm{N}, \hat{T})$ *is a bi-CCC with cartesian-closed structure inherited from* $\mathrm{Conc}(p, \mathrm{N})$ *and coproducts as in Lemma 4.4(2), the forgetful functor* $\mathrm{U} : \mathrm{ConcCl}(p, \mathrm{N}, \hat{T}) \to \mathbb{A}$ *strictly preserves finite products and finite coproducts, and one obtains the diagram below (cf. diagrams (4) and (5)):*

$$\begin{tikzcd}
\mathrm{ConcCl}(p, \mathrm{N}, \hat{T}) \arrow[loop left, "\mathrm{H}\hat{T}k"] \arrow[r, hook, "\perp"'] & \mathrm{Conc}(p, \mathrm{N}) \arrow[l, bend right] \arrow[loop above, "\mathrm{H}\hat{T}j"] \arrow[r, hook, "\perp"] & \mathrm{Krip}(p, \mathrm{N}) \arrow[l, bend right, "\mathrm{K}"'] \arrow[l, bend left, "\mathrm{H}", "\perp"'] \arrow[loop right, "\hat{T}"]
\end{tikzcd} \tag{11}$$

We now refine the abstract OHR construction for $\lambda_{\mathrm{c}}(\mathrm{S})$ (Sec. 7) to incorporate sums. Consider a fixed $\lambda_{\mathrm{c}}^{+}(\mathrm{S})$-model $(\mathcal{M}, T, s)$ and an indexing set $\mathbb{I}$ such that for each $i \in \mathbb{I}$ one has chosen data as in (6), and make the assumptions of Assump. 7.1. For each $i \in \mathbb{I}$ one now obtains *three* categories: $\mathrm{Krip}_{\mathcal{M}, F_i} := \mathrm{Krip}(\mathrm{cod}, \mathrm{N}_{F_i})$, $\mathrm{Conc}_{\mathcal{M}, F_i} := \mathrm{Conc}(\mathrm{cod}, \mathrm{N}_{F_i})$ and $\mathrm{ConcCl}_{\mathcal{M}, F_i, \hat{T}_i} := \mathrm{ConcCl}(\mathrm{cod}, \mathrm{N}_{F_i}, \hat{T}_i)$. Similarly, taking the fibration for logical relations $\prod_{i \in \mathbb{I}} \mathrm{cod}$ one obtains a category $\mathcal{K} := \mathrm{Krip}\big(\prod_{i \in \mathbb{I}} \mathrm{cod}, \langle \mathrm{N}_{F_i} \rangle_{i \in \mathbb{I}}\big)$ as well as subcategories $\mathcal{C} := \mathrm{Conc}\big(\prod_{i \in \mathbb{I}} \mathrm{cod}, \langle \mathrm{N}_{F_i} \rangle_{i \in \mathbb{I}}\big)$ and $\mathcal{CCl} := \mathrm{ConcCl}\big(\prod_{i \in \mathbb{I}} \mathrm{cod}, \langle \mathrm{N}_{F_i} \rangle_{i \in \mathbb{I}}, \hat{T}\big)$, for $\hat{T}$ the monad defined in Lemma 7.1.

Because cartesian liftings are determined component-wise (Lemma 7.1(1)), so is $\hat{T}$-closure. It follows that all the properties of Lemma 7.1 extend to $\mathcal{CCl}$. Moreover, $\mathcal{CCl}$ acquires a bi-CCC structure and a strong monad W with underlying functor $\mathrm{H}\hat{T}k$ by Lemma 10.1. We call $(\mathcal{CCl}, \mathrm{W}, \hat{s})$ the *abstract OHR model* on $(\mathcal{M}, T, s)$. The situation is summarised below (*cf.* Figure 1b and diagram (11)):

$$\begin{tikzcd}
\mathrm{ConcCl}_{\mathcal{M}, F_i, \hat{T}_i} \arrow[r, hook, "\perp"] & \mathrm{Conc}_{\mathcal{M}, F_i} \arrow[l, bend right] \arrow[r, hook, "\perp", "j"'] & \mathrm{Krip}_{\mathcal{M}, F_i} \arrow[l, bend right]
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{CCl} \arrow[loop left, "\mathrm{W} := \mathrm{H}\hat{T}k"] \arrow[r, hook, "\perp"] & \mathcal{C} \arrow[l, bend right] \arrow[r, hook, "\perp"] & \mathcal{K} \arrow[l, bend right] \arrow[l, bend left, "\perp", "\mathrm{H}"'] \arrow[loop right, "\hat{T}"]
\end{tikzcd}$$

Next we consider logical relations over $\mathcal{CCl}$ (*cf.* diagram (7)). Extending Lemmas 7.2 and 7.3 to incorporate sums, one obtains the following extension of Prop. 7.1 with sum types.

Proposition 10.1. *Let $\mathcal{R} = \{\mathcal{R}_\sigma\}_{\sigma \in \mathbf{Ty}^+}$ be a hungry, $(\lambda, +, 0)$-compatible $\lambda_c^+(\mathrm{S})$-logical relation over $\mathrm{W}^{\top\top[\mathcal{R}]}$ and suppose $i_0 \in \mathbb{I}$ is such that $\hat{T}_{i_0} = T^{\top\top[\langle\mathcal{R}\rangle]}$ and $\overline{\hat{s}\llbracket\sigma\rrbracket}(i_0)(\beta) = \mathcal{R}^{\mathrm{val}}_\beta$ for all $\beta \in \mathbf{B}$. Then $\overline{\hat{s}\llbracket\sigma\rrbracket}(i_0) = \mathcal{R}^{\mathrm{val}}_\sigma$ and $(\overline{\mathrm{W}\hat{s}\llbracket\sigma\rrbracket})(i_0) = \mathcal{R}_\sigma$ for all $\sigma \in \mathbf{Ty}^+$, so every $f : \hat{s}\llbracket\Gamma\rrbracket \to \mathrm{W}\hat{s}\llbracket\sigma\rrbracket$ in CCl satisfies $\mathcal{R}$.*

It remains to instantiate the abstract construction. The development goes through verbatim, except in the definition of $\mathbb{I}$ (equation (8)) we must now assume that $\mathrm{Interp}(\mathbb{A}, F, \hat{T})$ consists of maps $r : \mathbf{B} \to \mathrm{ConcCl}_{\mathcal{M},F,\hat{T}}$ rather than $\mathbf{B} \to \mathrm{Conc}_{\mathcal{M},F}$. We therefore obtain an *OHR model* as in Def. 8.1. In this setting, Prop. 8.1 and Cor. 8.1 are as follows.

Theorem 10.1. *Let $\mathcal{R} = \{\mathcal{R}_\sigma\}_{\sigma \in \mathbf{Ty}}$ be a hungry, $(\lambda, +, 0)$-compatible $\lambda_c^+(\mathrm{S})$-logical relation over $\hat{\mathrm{W}} := \mathrm{W}^{\top\top[\mathcal{R}]}$ such that every global element $g : 1 \to s\llbracket\beta\rrbracket$ in $\mathcal{M}$ is in $\mathcal{R}^{\mathrm{val}}_\beta(\diamond)$. Then every $\textsc{ohr}(\mathcal{M})$-morphism $f : \hat{s}\llbracket\Gamma\rrbracket \to \mathrm{W}\hat{s}\llbracket\sigma\rrbracket$ satisfies $\mathcal{R}$.*

Corollary 10.1. *If $F : (\mathcal{N}, S, t) \to (\textsc{ohr}(\mathcal{M}), \mathrm{W}, \hat{s})$ is a $\lambda_c^+(\mathrm{S})$-morphism with a surjective map $(\mathrm{U} \circ F)_{1,t\llbracket\beta\rrbracket} : \mathcal{N}(1, t\llbracket\beta\rrbracket) \to \mathcal{M}(1, s\llbracket\beta\rrbracket)$, then any $f : \hat{s}\llbracket\Gamma\rrbracket \to \mathrm{W}\hat{s}\llbracket\sigma\rrbracket$ in $\textsc{ohr}(\mathcal{M})$ is in F's image.*

Since Lemma 8.2 extends to sum types without difficulty, we also recover full abstraction.

Theorem 10.2. *Let $(\mathcal{M}, T, s)$ be a small, well-pointed $\lambda_c^+(\mathrm{S})$-model such that for every $\beta \in \mathbf{B}$ and $g : 1 \to s\llbracket\beta\rrbracket$ the composite $\eta_{s\llbracket\beta\rrbracket} \circ g$ is definable. Then, if $\mathrm{Krip}(\prod_i \mathrm{cod}, \langle F\rangle_{i\in\mathbb{I}})$ admits a tractable hull functor and the counit of the adjunction $j \dashv \mathrm{H}$ is component-wise monic, the induced model $(\textsc{ohr}(\mathcal{M}), \mathrm{W}, \hat{s})$ is well-pointed and fully complete, hence fully abstract.*

11 EXAMPLES: FULLY ABSTRACT MODELS OVER Diff AND QBS

In this final section we give two examples of the OHR construction with sums. First we study a simple language for probabilistic programming. This is the fragment of the idealised Anglican language of Staton et al. [2016] without recursive types and constructors; for the semantics we use the category of *quasi-Borel spaces* [Heunen et al. 2017]. Then we turn to the language for automatic differentiation studied by Huot et al. [2020], extended with a global memory cell that may be read and written to. The semantics takes place in the category of *diffeological spaces* (*e.g.* Iglesias-Zemmour [2013]). In each case the hull functor is defined as in Remark 6.1, so tractable.

11.1 Probability over QBS

The category of *quasi-Borel spaces* was introduced by Heunen et al. [2017] as a setting for the denotational semantics of higher-order probabilistic programming languages. Naively, one would hope to interpret such languages in the category **Meas** of measurable spaces and measurable maps using the *Giry monad* [Giry 1982]. However, **Meas** is not cartesian closed; **QBS** rectifies this deficiency. **QBS** acts as a kind of conservative extension of the category of standard Borel spaces: there is a functor $R : \mathbf{Meas} \to \mathbf{QBS}$ and, if X, Y are standard Borel spaces, then $\mathbf{QBS}(RX, RY) = \mathbf{Meas}(X, Y)$ (see *e.g.* Ścibior et al. [2018]). Moreover, the Giry monad restricts to a monad P on **QBS**, and if X, Y are standard Borel spaces then maps $RX \to \mathrm{P}(RY)$ correspond to the *s-finite kernels* used by Staton [2017] to give a complete semantics for a first-order probabilistic language.

The signature $\mathrm{S}_{\mathrm{prob}}$ for our idealised probabilistic programming language is given in the box below. The type real represents the real numbers. To get a primitive representing each measurable function we use the (co)cartesian structure of **Meas**: where $m : \{\mathrm{real}\} \to \mathbf{Meas}$ interprets real as the standard Borel space $(\mathbb{R}, \Sigma_{\mathbb{R}})$, one obtains a measurable space $m\llbracket\gamma\rrbracket$ for all ground types γ. Constant maps and all the usual distributions are measurable; hence for any $r \in \mathbb{R}$ one has $\underline{\mathrm{const}_r} : 1 \to \mathrm{real}$.

base types: real; **primitives:** $\underline{f} : \gamma \to \nu$ for every $f : m\llbracket\gamma\rrbracket \to m\llbracket\nu\rrbracket$;
operations (for $\gamma \in \mathbf{G}$): $\mathrm{score} : \mathrm{real} \rightsquigarrow 1$, $\mathrm{sample}_\gamma : \gamma \rightsquigarrow \gamma$, $\mathrm{normalise}_\gamma : \gamma \rightsquigarrow \mathrm{real} * \gamma + 1 + 1$;

Semantically, we set $s(\mathrm{real})$ to be the quasi-Borel space corresponding to $(\mathbb{R}, \Sigma_{\mathbb{R}})$. Because $\mathbb{R}$ is standard Borel, $m[\![\gamma]\!]$ is standard Borel for every ground type γ. For each operation $\mathrm{op} : \gamma \rightsquigarrow \nu$ we therefore take $s[\![\mathrm{op}\,]\!]$ to be the Kleisli arrow corresponding to the s-finite kernel interpretation of that operation given by Staton [2017]. Finally, for a primitive $\underline{f} : \gamma \to \nu$ we use the fact that $\mathbf{QBS}(s[\![\gamma]\!], s[\![\nu]\!]) = \mathbf{Meas}(m[\![\gamma]\!], m[\![\nu]\!])$ to define $s[\![\underline{f}]\!] := \lambda(* \in 1)\,.\,f : 1 \to (s[\![\gamma]\!] \Rightarrow s[\![\nu]\!])$. Thus, we have a $\lambda_{\mathrm{c}}^{+}(\mathrm{S}_{\mathrm{prob}})$-model $(\mathbf{QBS}, \mathrm{P}, s)$ (we silently identify **QBS** with a suitable small subcategory).

The induced OHR model is fully abstract, since for every $g : 1 \to s[\![\mathrm{real}]\!]$ the composite $\eta^{\mathrm{P}}_{s[\![\mathrm{real}]\!]} \circ g$ is definable. Indeed, global elements in **QBS** are in bijective correspondence with global elements in **Set**, so we can identify g with a constant map const_r for some $r \in \mathbb{R}$. Then $\eta^{\mathrm{P}}_{s[\![\mathrm{real}]\!]} \circ g = s[\![\diamond \vdash \underline{\mathrm{const}_r}\,() : \mathrm{real}]\!]$, as required. So the OHR model over $(\mathbf{QBS}, \mathrm{P}, s)$ exists and is fully abstract.

The forgetful functor $\mathbf{QBS} \to \mathbf{Set}$ strictly preserves products and coproducts and evaluation in **QBS** is as in **Set**, so the relationship between $\textsc{ohr}(\mathbf{QBS})$ and **QBS** is very similar to that between $\textsc{ohr}(\mathbf{Fin})$ and **Fin** outlined in Sec. 9. The monad W is a restriction of P, products in $\textsc{ohr}(\mathbf{QBS})$ are as in **QBS**—hence as in **Set**—and the evaluation map is a restriction of that in **Set** (*cf.* (9)).

11.2 Global State over Diff

Just as quasi-Borel spaces were introduced to deal with the fact that **Meas** is not cartesian closed, so the category **Diff** of diffeological spaces and smooth maps was used by Huot et al. [2020] to deal with the fact that the usual setting for differential geometry, namely the category of cartesian spaces and smooth functions, is not cartesian closed. A *diffeological space* is a pair $(X, \mathcal{P}_X)$ consisting of a set X equipped with a set of *plots* $\mathcal{P}^U_X \subseteq \mathbf{Set}(U, X)$ for every $n \in \mathbb{N}$ and open subset $U \subseteq \mathbb{R}^n$, subject to certain axioms. One then widens the definition of smooth function from differential geometry to this clean axiomatic setting: one calls a function *smooth* if it send plots to plots.

Huot et al. [2020] show that **Diff** is a natural setting for studying the denotational semantics of syntactic (forward mode) automatic differentiation for neural network programming. We consider an extension of their simple language for AD with a single memory cell. One may *lookup* the value of the cell or *update* it to a new shared value; we assume values range over a fixed set V. The relevant signature S_{GS} is defined in the box below, where the primitives + and × represent the usual operations on $\mathbb{R}$ and ς represents the sigmoid function $\varsigma(x) = (1 + e^{-x})^{-1}$.

base types: real, Val; **operations:** $\mathrm{lookup} : 1 \rightsquigarrow \mathrm{Val}$; $\mathrm{update} : \mathrm{Val} \rightsquigarrow 1$;
primitives: $\underline{r} : \mathrm{real}$ (for $r \in \mathbb{R}$); $+, \times : \mathrm{real} * \mathrm{real} \to \mathrm{real}$; $\varsigma : \mathrm{real} \to \mathrm{real}$; $\underline{v} : \mathrm{Val}$ (for $v \in V$).

For the semantic interpretation, set $s(\mathrm{real}) := (\mathbb{R}, \mathcal{P}_{\mathbb{R}})$, where $\mathcal{P}^U_{\mathbb{R}}$ is the set of smooth maps $U \to \mathbb{R}$. We interpret Val as a *coarse* diffeological space: $s(\mathrm{Val}) := (V, \mathcal{D}_V)$ where $\mathcal{D}^U_Y := \mathbf{Set}(U, Y)$. The memory cell is modelled by the *global state monad* $G_V(X) := V \Rightarrow (V \times X)$ on **Diff**. Primitives are interpreted by the corresponding (smooth) functions in **Set**; the usual set maps interpreting lookup and update (*e.g.* [Kammar 2014]) are smooth because $s(\mathrm{Val})$ is coarse. This defines a $\lambda_{\mathrm{c}}^{+}(\mathrm{S}_{\mathrm{GS}})$-model $(\mathbf{Diff}, G_V, s)$. The global elements in **Diff** are exactly the global elements in **Set** so, identifying **Diff** with a suitable small subcategory, the OHR model over $(\mathbf{Diff}, G_S, s)$ exists and is fully abstract.

The forgetful functor $\mathbf{Diff} \to \mathbf{Set}$ strictly preserves products and coproducts, and evaluation in **Diff** is evaluation in **Set**. Thus, just as for the **QBS** example, the relationship between the OHR model and $(\mathbf{Diff}, G_S, s)$ is very similar to that between $\textsc{ohr}(\mathbf{Fin})$ and **Fin** outlined in Sec. 9.

ACKNOWLEDGMENTS

We thank the authors of [Matache et al. 2021] for helpful discussions relating their work to ours, C. Matache & S. Staton for the models in Exs. 1.1 and 1.2, and D. McDermott, H. Paquet, N. Arkor and P. B. Levy for insightful comments. Finally we thank the reviewers for their thoughtful suggestions.

REFERENCES

S. Abramsky, K. Honda, and G. McCusker. 1998. A fully abstract game semantics for general references. In *Proceedings of the Thirteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No.98CB36226)*. IEEE Comput. Soc. https://doi.org/10.1109/lics.1998.705669

S. Abramsky and R. Jagadeesan. 1994. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic* 59, 2 (June 1994), 543–574. https://doi.org/10.2307/2275407

S. Abramsky, R. Jagadeesan, and P. Malacaria. 2000. Full Abstraction for PCF. *Information and Computation* 163, 2 (Dec. 2000), 409–470. https://doi.org/10.1006/inco.2000.2930

J. Adamek, H. Herrlich, and G.E. Strecker. 2009. *Abstract and Concrete Categories: The Joy of Cats*. Dover Publications. https://books.google.co.jp/books?id=rqT4PgAACAAJ

M. Alimohamed. 1995. A Characterization of Lambda Definability in Categorical Models of Implicit Polymorphism. *Theor. Comput. Sci.* 146, 1-2 (July 1995), 5–23. https://doi.org/10.1016/0304-3975(94)00283-O

F. Borceux. 1994. *Handbook of Categorical Algebra, volume 1*. Cambridge University Press. https://doi.org/10.1017/cbo9780511525858

R. Cartwright, P.L. Curien, and M. Felleisen. 1994. Fully Abstract Semantics for Observably Sequential Languages. 111, 2 (jun 1994), 297–401. https://doi.org/10.1006/inco.1994.1047

P. Clairambault and M. de Visme. 2020. Full abstraction for the quantum lambda-calculus. *Proceedings of the ACM on Programming Languages* 4, POPL (jan 2020), 1–28. https://doi.org/10.1145/3371131

P.-L. Curien. 2007. Definability and Full Abstraction. *Electronic Notes in Theoretical Computer Science* 172 (April 2007), 301–310. https://doi.org/10.1016/j.entcs.2007.02.011

U. de' Liguoro. 1996. *PCF Definability via Kripke Logical Relations (after O'Hearn and Riecke)*. Technical Report. Laboratoire d'Informatique de l'Ecole Normale Supérieure.

T. Ehrhard, C. Tasson, and M. Pagani. 2014. Probabilistic coherence spaces are fully abstract for probabilistic PCF. In *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. ACM. https://doi.org/10.1145/2535838.2535865

M. Fiore, A. Jung, E. Moggi, P. O'Hearn, Riecke J., G. Rosolini, and I. Stark. 1996. Domains and denotational semantics: History, accomplishments and open problems. *Bulletin of EATCS* 59 (1996), 227–256. Edited by A. Jung.

M. Fiore, G. Plotkin, and D. Turi. 1999. Abstract Syntax and Variable Binding. In *Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science (LICS '99)*. IEEE Computer Society, Washington, DC, USA, 193–. http://dl.acm.org/citation.cfm?id=788021.788948

M. Fiore and A. Simpson. 1999. Lambda Definability with Sums via Grothendieck Logical Relations. In *Typed Lambda Calculi and Applications*, J.-Y. Girard (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 147–161.

P. Freyd. 1972. Aspects of topoi. *Bulletin of the Australian Mathematical Society* 7, 1 (aug 1972), 1–76. https://doi.org/10.1017/s0004972700044828

J. Y. Girard. 1989. *Proofs and types*. Cambridge University Press, Cambridge.

M. Giry. 1982. A categorical approach to probability theory. In *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 68–85. https://doi.org/10.1007/bfb0092872

J. Goubault-Larrecq, S. Lasota, and D. Nowak. 2008. Logical relations for monadic types. *Mathematical Structures in Computer Science* 18, 06 (Oct. 2008), 1169. https://doi.org/10.1017/s0960129508007172

J. Goubault-Larrecq, S. Lasota, D. Nowak, and Y. Zhang. 2004. Complete Lax Logical Relations for Cryptographic Lambda-Calculi. In *Computer Science Logic*. Springer Berlin Heidelberg, 400–414. https://doi.org/10.1007/978-3-540-30124-0_31

C. A. Hermida. 1993. *Fibrations, Logical Predicates and Indeterminates*. Ph.D. Dissertation. University of Edinburgh.

C. Heunen, O. Kammar, S. Staton, and H. Yang. 2017. A Convenient Category for Higher-Order Probability Theory. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science* (Reykjavík, Iceland) *(LICS '17)*. IEEE Press, Article 77, 12 pages.

M. Huot, S. Staton, and M. Vákár. 2020. Correctness of Automatic Differentiation via Diffeologies and Categorical Gluing. In *Lecture Notes in Computer Science*. Springer International Publishing, 319–338. https://doi.org/10.1007/978-3-030-45231-5_17

J.M.E. Hyland and C.-H.L. Ong. 2000. On Full Abstraction for PCF: I, II, and III. *Information and Computation* 163, 2 (Dec. 2000), 285–408. https://doi.org/10.1006/inco.2000.2917

P. Iglesias-Zemmour. 2013. *Diffeology*. American Mathematical Society, Providence, Rhode Island.

B. Jacobs. 1993. Comprehension categories and the semantics of type dependency. *Theoretical Computer Science* 107, 2 (Jan. 1993), 169–207. https://doi.org/10.1016/0304-3975(93)90169-t

B. Jacobs. 1999. *Categorical Logic and Type Theory*. Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam.

P. T. Johnstone. 2002. *Sketches of an elephant : a topos theory compendium*. Oxford University Press, Oxford New York.

A. Jung and J. Tiuryn. 1993. A new characterization of lambda definability. In *Typed Lambda Calculi and Applications*, Marc Bezem and Jan Friso Groote (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 245–257.

O. Kammar. 2014. *An Algebraic Theory of Type-and-Effect Systems.* Ph.D. Dissertation. University of Edinburgh.

O. Kammar and D. McDermott. 2018. Factorisation Systems for Logical Relations and Monadic Lifting in Type-and-effect System Semantics. *Electronic Notes in Theoretical Computer Science* 341 (2018), 239 – 260. https://doi.org/10.1016/j.entcs.2018.11.012 Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIV).

O. Kammar and G. D. Plotkin. 2012. Algebraic foundations for effect-dependent optimisations. In *Proceedings of the 39th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '12*. ACM Press. https://doi.org/10.1145/2103656.2103698

S. Katsumata. 2005. A Semantic Formulation of ⊤⊤-Lifting and Logical Predicates for Computational Metalanguage. In *Computer Science Logic*. Springer Berlin Heidelberg, 87–102. https://doi.org/10.1007/11538363_8

S. Katsumata. 2008. A Characterisation of Lambda Definability with Sums Via ⊤⊤-Closure Operators. In *Computer Science Logic*, Michael Kaminski and Simone Martini (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 278–292.

S. Katsumata. 2013. Relating computational effects by ⊤⊤-lifting. *Information and Computation* 222 (2013), 228 – 246. https://doi.org/10.1016/j.ic.2012.10.014 38th International Colloquium on Automata, Languages and Programming (ICALP 2011).

A. Kock. 1972. Strong functors and monoidal monads. *Archiv der Mathematik* 23, 1 (dec 1972), 113–120. https://doi.org/10.1007/bf01304852

S. Lasota, D. Nowak, and Y. Zhang. 2007. On Completeness of Logical Relations for Monadic Types. In *Advances in Computer Science - ASIAN 2006. Secure Software and Related Issues*. Springer Berlin Heidelberg, 223–230. https://doi.org/10.1007/978-3-540-77505-8_17

F. W. Lawvere. 2006. Diagonal Arguments and Cartesian Closed Categories. *Reprints in Theory and Applications of Categories* 15 (2006), 1–13. http://www.tac.mta.ca/tac/reprints/articles/15/tr15.pdf

F. Loregian and E. Riehl. 2020. Categorical notions of fibration. *Expositiones Mathematicae* 38, 4 (Dec. 2020), 496–514. https://doi.org/10.1016/j.exmath.2019.02.004

Q. M. Ma and John C. Reynolds. 1992. Types, abstraction, and parametric polymorphism, part 2. In *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1–40. https://doi.org/10.1007/3-540-55511-0_1

M. Marz. 2000. A fully abstract model for sequential computation. *Electronic Notes in Theoretical Computer Science* 35 (2000), 133–152. https://doi.org/10.1016/s1571-0661(05)80735-2

C. Matache, S. Moss, , and S. Staton. 2021. Recursion and Sequentiality in Categories of Sheaves. *To appear in Proceedings of 6th International Conference on Formal Structures for Computation and Deduction (FSCD 2021)* (2021). https://doi.org/10.4230/LIPIcs.FSCD.2021.25

R. Milner. 1977. Fully abstract models of typed λ-calculi. *Theoretical Computer Science* 4, 1 (Feb. 1977), 1–22. https://doi.org/10.1016/0304-3975(77)90053-6

J. C. Mitchell and A. Scedrov. 1993. Notes on sconing and relators. In *Computer Science Logic*. Springer Berlin Heidelberg, 352–378. https://doi.org/10.1007/3-540-56992-8_21

E. Moggi. 1989. Computational Lambda-Calculus and Monads. In *Proceedings of the Fourth Annual Symposium on Logic in Computer Science* (Pacific Grove, California, USA). IEEE Press, 14–23.

E. Moggi. 1991. Notions of computation and monads. *Inf. Comput.* 93, 1 (1991), 55–92.

A. S. Murawski and N. Tzevelekos. 2012. Algorithmic Games for Full Ground References. In *Automata, Languages, and Programming*. Springer Berlin Heidelberg, 312–324. https://doi.org/10.1007/978-3-642-31585-5_30

P. W. O'Hearn and J. G. Riecke. 1995. Kripke Logical Relations and PCF. *Information and Computation* 120, 1 (1995), 107 – 116. https://doi.org/10.1006/inco.1995.1103

B. Pareigis. 1977. Non-additive ring and module theory II: C-categories, C-functors and C-morphisms. *Publicationes mathematicae* 24, 3 (January 1977).

G. D. Plotkin. 1973. *Lambda-definability and logical relations.* Technical Report. University of Edinburgh.

G. D. Plotkin. 1977. LCF considered as a programming language. *Theoretical Computer Science* 5, 3 (Dec. 1977), 223–255. https://doi.org/10.1016/0304-3975(77)90044-5

G. D. Plotkin. 1980. Lambda-Definability in the Full Type Hierarchy. In *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, Jonathan P. Seldin and J. Roger Hindley (Eds.). Academic Press.

G. D. Plotkin and J. Power. 2003. Algebraic Operations and Generic Effects. *Applied Categorical Structures* 11, 1 (2003), 69–94. https://doi.org/10.1023/a:1023064908962

J. Power and E. Robinson. 2000. Logical Relations and Data Abstraction. In *Computer Science Logic*. Springer Berlin Heidelberg, 497–511. https://doi.org/10.1007/3-540-44622-2_34

J. G. Riecke and A. Sandholm. 2002. A Relational Account of Call-by-Value Sequentiality. *Information and Computation* 179, 2 (Dec. 2002), 296–331. https://doi.org/10.1006/inco.2002.2957

G. Scherer. 2017. Deciding Equivalence with Sums and the Empty Type. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages* (Paris, France) *(POPL 2017)*. Association for Computing Machinery, New York, NY, USA, 374–386. https://doi.org/10.1145/3009837.3009901

A. Ścibior, O. Kammar, V. Vákár, S. Staton, H. Yang, Y. Cai, K. Ostermann, S. K. Moss, C. Heunen, and Z. Ghahramani. 2018. Denotational validation of higher-order Bayesian inference. *Proceedings of the ACM on Programming Languages* 2, POPL (Jan. 2018), 1–29. https://doi.org/10.1145/3158148

J. M. Souriau. 1980. Groupes differentiels. In *Differential Geometrical Methods in Mathematical Physics*, P. L. García, A. Pérez-Rendón, and J. M. Souriau (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 91–128.

S. Staton. 2017. Commutative Semantics for Probabilistic Programming. In *Programming Languages and Systems*. Springer Berlin Heidelberg, 855–879. https://doi.org/10.1007/978-3-662-54434-1_32

S. Staton, H. Yang, F. Wood, C. Heunen, and O. Kammar. 2016. Semantics for probabilistic programming. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*. ACM. https://doi.org/10.1145/2933575.2935313

R. Street. 1972. The formal theory of monads. *Journal of Pure and Applied Algebra* 2, 2 (1972), 149 –168. https://doi.org/10.1016/0022-4049(72)90019-9