Categorical Models of Parametric Polymorphism

Lars Birkedal
IT University of Copenhagen
birkedal@itu.dk

Rasmus E. Møgelberg
IT University of Copenhagen
mogel@itu.dk

Abstract

We propose a new category theoretic formulation of relational parametricity based on a logic for reasoning about parametricity given by Abadi and Plotkin [11]. The logic can be used to reason about parametric models, such that we may prove consequences of parametricity that to our knowledge have not been proved before for existing category theoretic notions of relational parametricity. We provide examples of parametric models and we describe a way of constructing parametric models from given models of the second-order lambda calculus.

1. Introduction

The notion of parametricity for models of polymorphic type theories intuitively states that a function of polymorphic type behaves the same way on all type instances. Reynolds [12] discovered that parametricity is central for modelling data abstraction and proving representation independence results. The idea is that a client of an abstract data type is modelled as a polymorphic function; parametricity then guarentees that the client cannot distinguish between different implementations of the abstract data type. Reynolds also observed that parametricity can be used for encoding (inductive and coinductive) data types. See [17, 8] for expository introductions.

In 1983 Reynolds gave a precise formulation of parametricity for set-theoretic models, called relational parametricity [12]. It basically states that a term of polymorphic type preserves relations between types: if term u has type $\prod\alpha\colon \mathsf{Type}.\, \sigma$ and $R \subset \tau \times \tau'$ is a relation, then

$$u(\tau)(\sigma[R])u(\tau'),$$

where $\sigma[R]$ is a relational interpretation of the type σ defined inductively over the structure of σ. Equivalently, parametricity could be defined as the identity extension property: for all terms u, v of type $\sigma(\vec{\alpha})$,

$$u(\sigma[e\vec{q}_\alpha])v \iff u = v.$$

However, Reynolds himself later proved that set-theoretic models do not exist [13]. In 1992 Ma and Reynolds [7] then gave a new formulation of parametricity phrased in terms of more general models (PL-categories of Seely [16]). One may formulate Ma and Reynolds' notion in the language of λ_2-fibrations[1] as follows. A λ_2-fibration $E \to B$ is parametric with respect to a given logic on E if there exists a reflexive graph of λ_2-fibrations, where the restriction to the fibers over the terminal object is the reflexive graph

$$E_1 \rightleftarrows \mathbf{LR}(E_1)$$

of logical relations with domain, codomain maps and the middle map mapping a type to the identity on that type. (See [7, 5] for more details.)

In recent work by Birkedal and Rosolini on parametric domain-theoretic models it became clear that this is not the right categorical formulation of parametricity: it appears that the definition does not allow one to prove the expected consequences of parametricity such as data abstraction and the encoding of data types. Indeed, these consequences have only been proved for specific models, see, e.g., [17, 3], using specific properties of the models.

In this article we propose a new category-theoretic formulation of parametricity, called a *parametric APL-structure*, which *does* allow one to prove the expected properties of parametricity in general. Our basis is a logic for reasoning about parametricity given by Abadi and Plotkin [11]. In this logic one can formulate parametricity as a schema and prove the expected consequences of parametricity. An APL-structure is a category-theoretic model of Abadi and Plotkin's logic, for which we prove soundness and completeness, thereby answering a question posed in [11, Page 5]. Each APL-structure contains a model of the second-order lambda calculus, which we may reason about using the logic.

We also provide a completion process that given an internal model of λ_2 (see [4, 14]) produces a parametric APL-structure. The λ_2-fibration of this APL-structure is essen-

[1] A λ_2-fibration is a fibration with enough properties to model second-order lambda calculus, see section 3

tially the one obtained in [14] and thus we prove that the models obtained in [14] in fact satisfy the consequences of parametricity (as expected, but not shown in the literature before).

The consequences of parametricity proved earlier for specific models [3, 17, 2] all seem to use well-pointedness, i.e., the property, that morphisms $f : A \to B$ are determined by their values on global elements $a : 1 \to A$, and, indeed, in his recent Ph.D. thesis, Dunphy [2] reaches the conclusion that parametricity is only useful in connection with well-pointedness. For parametric APL-structures, we do not need to use well-pointedness to prove the expected consequences of parametricity. Loosely speaking, the point is that our notion of parametric APL-structure includes an appropriate extensional logic to reason with. In *loc. cit.*, the ambient world of set theory is used as the logic and thus extensionality there amounts to asking for well-pointedness. We provide a family of concrete parametric APL-structures, including non-well pointed ones. Thus parametricity *is* useful for proving consequences also for non-well-pointed models.

In Section 2, we recall Abadi and Plotkin's logic. The reader is warned that our version of the logic is a little different from the one described in [11]. In Section 3 we define the notion of an APL-structure and prove soundness and completeness with respect to Abadi and Plotkin's logic. In Section 4 we define *parametric* APL-structures and show show how to use the logic to reason about them. Section 5 contains a family of parametric APL-structures. Finally, Section 6 contains a discussion of the differences of Ma & Reynolds' formulation of parametricity and the one given here, and Section 7 describes the parametric completion process. We conclude in Section 8.

Due to space constraints, many details and proofs have been omitted from this article. For these, we refer to the full technical report [9].

2 Abadi and Plotkin's Logic

Abadi and Plotkin's logic for parametricity is based on the second-order lambda calculus (λ_2). We shall first establish notation for λ_2. Expressions are on the form

$$\alpha_1 \colon \mathsf{Type}, \dots, \alpha_n \colon \mathsf{Type} \mid x_1 \colon \sigma_1, \dots, x_m \colon \sigma_m \vdash M \colon \sigma.$$

The α's are the free type variables and the collection of α's is called the kind context. The x's are the free variables and the collection $x_1 \colon \sigma_1, \dots, x_m \colon \sigma_m$ is called the type context. The grammar for types is given by

$$\sigma ::= \alpha \mid \sigma \times \sigma' \mid \sigma \to \sigma' \mid \textstyle\prod \alpha \colon \mathsf{Type}.\, \sigma$$

and the grammar for terms is

$$\begin{array}{rcl} M & ::= & x \mid \langle M, N \rangle \mid \pi M \mid \pi' M \mid M\, N \mid \lambda x \colon \sigma. \\ & & M \mid M\, \sigma \mid \Lambda \alpha \colon \mathsf{Type}.\, M \end{array}$$

We also use s, t, u, f and g to range over terms. By external equality of terms we mean the smallest equivalence relation on terms defined by α-conversion, standard rules for products and the two versions of β, η-conversions. External equality of terms is denoted $=$.

Formulas of Abadi & Plotkin's logic are on the form

$$\vec{\alpha} \colon \mathsf{Type} \mid \vec{x} \colon \vec{\sigma} \mid R_1 \subset \tau_1 \times \tau_1', \dots, R_n \subset \tau_n \times \tau_n' \vdash \phi \colon \mathsf{Prop}.$$

The R's are free relational variables and they make up the relation context. For the relation context to be well-formed we require that all τ_i, τ_i' are well-formed types. In the following, we let Ξ range over kind contexts, Γ over type contexts and Θ over relation contexts, and thus write $\Xi \mid \Gamma \mid \Theta \vdash \phi$ for a formula.

The grammar for formulas is

$$\begin{array}{rcl} \phi & ::= & (M =_\sigma N) \mid R(M, N) \mid \bot \mid \top \mid \phi \wedge \psi \mid \\ & & \phi \vee \psi \mid \phi \supset \psi \mid \forall \alpha \colon \mathsf{Type}.\, \phi \mid \forall x \colon \sigma.\, \phi \mid \\ & & \forall R \subset \sigma \times \tau.\, \phi \mid \exists \alpha \colon \mathsf{Type}.\, \phi \mid \exists x \colon \sigma.\, \phi \mid \\ & & \exists R \subset \sigma \times \tau.\, \phi \mid \sigma[\vec{\rho}](s, t) \end{array}$$

where each ρ_i is a definable relation (to be discussed below). For M, N two terms of type σ in the same context we may form $M =_\sigma N$ called *internal equality of terms*. If $R \subset \tau \times \tau'$ is in the relation context, and $M \colon \tau$, $N \colon \tau'$ are terms then $R(M, N)$ is a formula. We may quantify over types, variables and relations.

Given a formula in a context with two free variables, we may abstract the free variables and obtain a *definable relation*. To be more precise, definable relations are constructed using the rule:

$$\frac{\Xi \mid \Gamma, x \colon \sigma, y \colon \tau \mid \Theta \vdash \phi \colon \mathsf{Prop}}{\Xi \mid \Gamma \mid \Theta \vdash (x \colon \sigma, y \colon \tau).\, \phi \subset \sigma \times \tau}$$

For definable relations $\rho = (x \colon \sigma, y \colon \tau).\, \phi \subset \sigma \times \tau$ and terms $M \colon \sigma$, $N \colon \tau$, we write $M \rho N$ or $\rho(M, N)$ for the proposition obtained by substituting M, N for x and y in ϕ.

The last construction in the grammar for formulas is a relational interpretation of types; we may substitute definable relations for free type variables in types and obtain relations:

$$\frac{\begin{array}{c} \alpha_1, \dots, \alpha_n \vdash \sigma \colon \mathsf{Type} \\ \Xi \mid \Gamma \mid \Theta \vdash \rho_1 \subset \tau_1 \times \tau_1', \dots, \rho_n \subset \tau_n \times \tau_n' \\ \Xi \mid \Gamma \vdash t \colon \sigma(\vec{\tau}/\vec{\alpha}) \quad \Xi \mid \Gamma \vdash s \colon \sigma(\vec{\tau}'/\vec{\alpha}) \end{array}}{\Xi \mid \Gamma \mid \Theta \vdash \sigma[\vec{\rho}](s, t)}$$

We shall refer to this construction as the *relational interpretation of types*. It is important to notice that this is an actual construction in the language. This differs from the logic defined in [11], where $\sigma[\vec{\rho}]$ is just short-hand notation for an operation of substitution defined inductively on the structure of σ. We would like to be able to reason about $\sigma[\vec{\rho}]$ also for type constants σ, so in our case induction will not work.

$$\frac{\vec{\alpha} \vdash \alpha_i \qquad \Xi \mid \Gamma \mid \Theta \vdash \vec{\rho} \subset \vec{\tau} \times \vec{\tau}'}{\Xi \mid \Gamma \mid \Theta \mid \top \vdash \alpha_i[\vec{\rho}] \equiv \rho_i} \tag{1}$$

$$\frac{\vec{\alpha} \vdash \sigma \to \sigma' \qquad \Xi \mid \Theta \vdash \vec{\rho} \subset \vec{\tau} \times \vec{\tau}'}{\Xi \mid \Gamma \mid \Theta \mid \top \vdash (\sigma \to \sigma')[\vec{\rho}] \equiv (\sigma[\vec{\rho}] \to \sigma'[\vec{\rho}])} \tag{2}$$

$$\frac{\vec{\alpha} \vdash \prod \beta.\, \sigma(\vec{\alpha}, \beta) \qquad \Xi \mid \Theta \vdash \vec{\rho} \subset \vec{\tau} \times \vec{\tau}'}{\begin{array}{ll}\Xi \mid \Gamma \mid \Theta \mid \top \vdash & (\prod \beta.\, \sigma(\vec{\alpha}, \beta))[\vec{\rho}] \equiv \\ & \forall(\beta, \beta', R \subset \beta \times \beta').\, \sigma[\vec{\rho}, R])\end{array}} \tag{3}$$

$$\begin{aligned}&\forall \alpha, \beta \colon \mathsf{Type}. \forall x, x' \colon \alpha. \forall y, y' \colon \beta. \forall R \subset \alpha \times \beta. \\ &\quad R(x, y) \wedge x =_\alpha x' \wedge y =_\beta y' \supset R(x', y')\end{aligned} \tag{4}$$

Figure 1. Axioms for Abadi & Plotkin's logic

2.1 Axioms and Rules

We shall use the notation $\Xi \mid \Gamma \mid \Theta \mid \phi \vdash \psi$ meaning that the two propositions ϕ, ψ are well-formed in the context $\Xi \mid \Gamma \mid \Theta$ and that in this context ϕ implies ψ.

We assume the usual axioms and inference rules for propositional logic, the mate rules for quantification ([5, Section 4.1]), and that external equality implies internal equality. The axioms (1)-(3) reflect the inductive definition of $\sigma[\vec{\rho}]$ in [11]. We use the notation $\rho \equiv \rho'$ between definable relations of the same type for $\forall x, y.\, \rho(x, y) \supset\subset \rho'(x, y)$. We have also used the arrow notation on definable relations $\rho \subset \sigma \times \tau$ and $\rho' \subset \sigma' \times \tau'$ giving

$$(\rho \to \rho') \subset (\sigma \to \sigma') \times (\tau \to \tau')$$

defined as

$$f(\rho \to \rho')g = \forall x \colon \sigma \forall y \colon \tau (x \rho y \supset f x \rho' g y)$$

and we have used

$$\forall(\alpha, \beta, R \subset \alpha \times \beta).\, \rho \subset (\prod \alpha \colon \mathsf{Type}.\, \sigma) \times (\prod \alpha \colon \mathsf{Type}.\, \tau)$$

defined for $\Xi, \alpha, \beta \mid \Gamma \mid \Theta, R \subset \alpha \times \beta \vdash \rho \subset \sigma \times \tau$ well-formed and $\Xi \mid \Gamma \mid \Theta$ and $\Xi, \alpha \vdash \sigma \colon \mathsf{Type}$ and $\Xi, \alpha \vdash \tau \colon \mathsf{Type}$ as

$$\begin{gathered}t(\forall(\alpha, \beta, R \subset \alpha \times \beta).\, \rho)u = \\ \forall \alpha, \beta \colon \mathsf{Type}. \forall R \subset \alpha \times \beta.\, (t\alpha)\rho(u\beta)\end{gathered}$$

Axiom (4) is a substitution axiom for definable relations.

3 APL-structures

In this section we give a precise definition of an APL-structure, which can be used to interpret Abadi and Plotkin's logic. We use the definitions and results of Appendix A.

Let us first recall the definition of a λ_2-fibration

Definition 3.1. A fibration $E \to B$ is a λ_2-*fibration* if it is fibred cartesian closed, has a generic object $\Omega \in B$, products in B, and simple Ω-products, i.e., right adjoints $\prod_\pi$ to the reindexing functors π^* for projections $\pi \colon \Omega^{n+1} \to \Omega^n$, satisfying a Beck-Chevalley condition.

Definition 3.2. A *pre-APL-structure* consists of

1. Fibrations:

$$\begin{tikzcd} & \mathbf{Prop} \arrow[d, "r"] \\ \mathbf{Type} \arrow[r, hook, "I"] \arrow[dr, "p"'] & \mathbf{Ctx} \arrow[d, "q"] \\ & \mathbf{Kind} \end{tikzcd}$$

where

- p is a $\lambda 2$-fibration.
- q is a fibration with fibred products
- (r, q) is an indexed first-order logic fibration (Definition A.3) which has products and coproducts with respect to projections $\Omega^{n+1} \to \Omega^n$ in $\mathbf{Kind}$ (Definition A.4) where Ω is the generic object of p.
- I is a faithful product preserving map of fibrations.

2. a contravariant morphism of fibrations:

$$\begin{tikzcd} \mathbf{Type} \times_{\mathbf{Kind}} \mathbf{Type} \arrow[rr, "U"] \arrow[dr] & & \mathbf{Ctx} \arrow[dl] \\ & \mathbf{Kind} & \end{tikzcd}$$

3. a family of bijections

$$\Psi_\Xi : \mathrm{Hom}_{\mathbf{Ctx}_\Xi}(\xi, U(\sigma, \tau)) \to \mathrm{Obj}(\mathbf{Prop}_{\xi \times I(\sigma \times \tau)})$$

for σ and τ in $\mathbf{Type}_\Xi$ and ξ in $\mathbf{Ctx}_\Xi$, which

- is natural in the domain variable ξ
- commutes with reindexing functors; that is, if $\rho : \Xi' \to \Xi$ is a morphism in $\mathbf{Kind}$ and $u : \xi \to U(\sigma, \tau)$ is a morphism in $\mathbf{Ctx}_\Xi$, then

$$\Psi_{\Xi'}(\rho^*(u)) = (\bar{\rho})^*(\Psi_\Xi(u))$$

where $\bar{\rho}$ is the cartesian lift of ρ.

Notice that Ψ is only defined on vertical morphisms.

Remark 3.3. Item 3 in Definition 3.2 implies that $(U(1_\Xi, 1_\Xi))_{\Xi \in \mathbf{Kind}}$is an indexed family of generic objects (Definition A.1). If, on the other hand we have an indexed family of generic objects $(\Sigma_\Xi)_{\Xi\in\mathbf{Kind}}$ and $\mathbf{Ctx}$ is cartesian closed, then we may define U to be $\Sigma^{-\times-}$ and thereby get items 2 and 3 for free. In general, however, $\mathbf{Ctx}$ will not be cartesian closed.

We will show how to interpret the subset of Abadi and Plotkin's logic excluding the relational interpretation of types ($\sigma[\vec{\rho}]$) in a pre-APL structure. As is well-known, one can interpret λ_2 in λ_2-fibrations as follows. A type $\alpha_1 \dots \alpha_n \vdash \alpha_i$ is interpreted as the object of $\mathbf{Type}$ over Ω^n corresponding to the i'th projection $\Omega^n \to \Omega$. For a type $\alpha_1 \dots \alpha_n \vdash \sigma$, we have $[\![\prod \alpha_i.\, \sigma]\!] = \prod_\pi [\![\vec{\alpha} \vdash \sigma]\!]$, where π is the projection forgetting the i'th coordinate. Since each fiber of the λ_2-fibration is cartesian closed, we may interpret the constructions of the simply typed λ-calculus using fibrewise constructions.

If $\Xi, \alpha \mid \Gamma \vdash t\colon \tau$ is a term and $\Xi \vdash \Gamma$ is well-formed, then we may interpret the term $\Xi \mid \Gamma \vdash \Lambda\alpha.\, t\colon \prod\alpha.\,\tau$ as the morphism corresponding to $[\![\Xi, \alpha \mid \Gamma \vdash t\colon \tau]\!]$ under the adjunction $\pi^* \dashv \prod_\pi$.

To interpret $\Xi \mid \Gamma \vdash t\, \sigma$, notice that $[\![\Xi \vdash \sigma]\!]$ corresponds to a map $\widehat{[\![\Xi \vdash \sigma]\!]}\colon [\![\Xi]\!] \to \Omega$. The morphism $[\![\Xi \mid \Gamma \vdash t\colon \prod\alpha.\,\tau]\!]$ corresponds by the adjunction $\pi^* \dashv \prod_\pi$ to a morphism in the fiber over $[\![\Xi]\!] \times \Omega$. We reindex this morphism along $\langle id_{[\![\Xi]\!]}, \widehat{[\![\Xi \vdash \sigma]\!]}\rangle\colon [\![\Xi]\!] \to [\![\Xi]\!] \times \Omega$ to get $[\![\Xi \mid \Gamma \vdash t\, \sigma]\!]$.

We may interpret contexts of the logic in $\mathbf{Ctx}$ as

$$[\![\Xi \mid \vec{x}\colon \vec{\sigma} \mid \vec{R} \subset \vec{\tau} \times \vec{\tau}']\!] = I(\textstyle\prod_i [\![\sigma_i]\!]) \times \prod_i U([\![\tau_i]\!], [\![\tau_i']\!])$$

where the interpretation of types refers to the interpretation in $\mathbf{Type}$.

Since each restriction of $\mathbf{Prop} \to \mathbf{Ctx}$ to a fiber of $\mathbf{Ctx} \to \mathbf{Kind}$ is a first-order logic fibration, we may interpret equality, the usual propositional connectives and quantification with respect to variables and relations. To be more precise, we shall interpret the proposition

$$\Xi \mid \Gamma \mid \Theta \vdash t =_\tau s$$

using equality on the morphisms $I([\![t]\!]) \circ \pi, I([\![s]\!]) \circ \pi$ with $\pi : [\![\Xi \mid \Gamma \mid \Theta]\!] \to I([\![\Xi \mid \Gamma]\!])$ the projection as usual in first order logic fibrations [5] and $\forall x\colon \sigma.\, \phi$ is interpreted using the right adjoint to the projection

$$[\![\Xi \mid \Gamma, x\colon \sigma \mid \Theta]\!] \to [\![\Xi \mid \Gamma \mid \Theta]\!].$$

Quantification over types is interpreted using the simple products and simple coproducts of the composable fibration.

To interpret $R(t, s)$, the idea is that the bijection Ψ tells us that maps into the interpretation of $R \subset \tau \times \tau'$ the same as definable relations from τ to τ'. The idea is that the definable relation $(x, y).\, R(x, y)$ should be the one corresponding to the projection. So to interpret $R(t, s)$ we apply Ψ to the projection $[\![\Xi \mid \Gamma \mid \Theta, R \subset \tau \times \tau']\!] \to [\![\Xi \mid - \mid R \subset \tau \times \tau']\!]$ which results in a proposition in the context $[\![\Xi \mid \Gamma, x\colon \tau, y\colon \tau' \mid \Theta, R \subset \tau \times \tau']\!]$, which we then may reindex using the interpretations of t and s.

Proposition 3.4. *The interpretation of the subset of Abadi & Plotkin's logic without the relational interpretation of types is sound with respect to the axioms and rules of the logic except (1)- (3).*

Proof. Most of the axioms are proved as in the proof of soundness of the interpretation of first order logic in first order logic fibrations. Axiom (4) is easily provable from definitions. See [9] for details. □

With Proposition 3.4 at hand we may reason about a pre-APL structure using the subset of Abadi & Plotkin's logic as an internal language. We will use this below.

An APL-structure should be a pre-APL-structure in which we can interpret types as relations. To give this definition we first define a fibration $\mathbf{Relations} \to \mathbf{RelCtx}$. The objects of $\mathbf{RelCtx}$ are vectors

$$(\Xi, \Xi', \sigma_1, \sigma_1', \dots, \sigma_n, \sigma_n')$$

where (Ξ, Ξ') is a pair of objects in $\mathbf{Kind}$, each $\sigma_i \in \mathrm{Obj}(\mathbf{Type}_\Xi)$ and each $\sigma_i' \in \mathrm{Obj}(\mathbf{Type}_{\Xi'})$. We think of these as objects

$$[\![\Xi, \Xi' \mid \vec{R} \subset \vec{\sigma} \times \vec{\sigma}']\!]$$

in $\mathbf{Ctx}$. Morphisms of $\mathbf{RelCtx}$ are morphisms in $\mathbf{Ctx}$ projecting to products of morphisms in $\mathbf{Kind}$ between the respective pair of $\mathbf{Kind}$ objects. Objects of $\mathbf{Relations}_{(\Xi,\Xi',\sigma_1,\sigma_1',\dots,\sigma_n,\sigma_n')}$are triples consisting of two types $\tau \in \mathrm{Obj}(\mathbf{Type}_\Xi)$ and $\tau' \in \mathrm{Obj}(\mathbf{Type}_{\Xi'})$ and a proposition

$$\phi \in \mathbf{Prop}_{[\![\Xi,\Xi' \mid x\colon \tau, y\colon \tau \mid \vec{R} \subset \vec{\sigma} \times \vec{\sigma}']\!]}.$$

A morphism from such a triple $(\tau_0, \tau_0', \phi_0)$ to another triple $(\tau_1, \tau_1', \phi_1)$ in the same fiber is a pair of morphisms in $\mathbf{Type}$, $t\colon \tau_0 \to \tau_1$, $s\colon \tau_0' \to \tau_1'$, such that

$$\phi_0 \vdash (I(\pi^*(t) \times \pi'^*(s)) \times id_{[\![\Xi \mid \Theta]\!]})^* \phi_1,$$

where π, π' are the projections from $[\![\Xi, \Xi']\!]$ to $[\![\Xi]\!]$ and $[\![\Xi']\!]$, respectively. In the internal language this condition means $\forall x\colon \tau_0, y\colon \tau_0'.\, \phi_0(x, y) \supset \phi_1(t(x), s(y))$.

Lemma 3.5. *The fibration* $\mathbf{Relations} \to \mathbf{RelCtx}$ *is a λ_2-fibration and we have two maps of λ_2-fibrations*

$$\begin{tikzcd} \mathbf{Type} \arrow[d] & \mathbf{Relations} \arrow[l, shift right, "\partial_0"'] \arrow[l, shift left, "\partial_1"] \arrow[d] \\ \mathbf{Kind} & \mathbf{RelCtx} \arrow[l, shift right, "\partial_0"'] \arrow[l, shift left, "\partial_1"] \end{tikzcd} \quad (5)$$

mapping a relation to its domain and codomain respectively.

Proof. We will only sketch this proof, and we will work in the internal logic using Proposition 3.4. The generic object in $\mathbf{RelCtx}$ is $(\alpha, \beta, \alpha \vdash \alpha, \beta \vdash \beta)$ which we think of as $\alpha, \beta \mid R \subset \alpha \times \beta$. Products of $\mathbf{RelCtx}$ is given by composition of contexts, i.e.,

$$(\Xi_0; \Xi_0' \vdash \vec{R}_0 \subset \vec{\sigma}_0 \times \vec{\tau}_0) \times (\Xi_1; \Xi_1' \vdash \vec{R}_1 \subset \vec{\sigma}_1 \times \vec{\tau}_1) = \\ \Xi_0, \Xi_1; \Xi_0', \Xi_1' \vdash \vec{R}_0 \subset \vec{\sigma}_0 \times \vec{\tau}_0, \vec{R}_1 \subset \vec{\sigma}_1 \times \vec{\tau}_1$$

Since the objects of $\mathbf{Relations}$ are propositions with two free variables, we may think of these as definable relations. Now, the fibrewise exponential in $\mathbf{Relations}$ becomes $\rho \to \rho'$ and the simple product becomes $\forall(\alpha, \beta, R).\rho$, both as defined in section 2.1. ☐

Definition 3.6. An *APL-structure* is a pre-APL-structure with a given extension of (5) to a reflexive graph, i.e., a map of λ_2-fibrations J such that $\partial_0 J = \partial_1 J = id$.

Given a type $[\![\vec{\alpha} \vdash \sigma \colon \mathsf{Type}]\!]$ we may apply J to it. Since this type is an object of $\mathbf{Type}_{\Omega^n}$ we get an object over the n'th power of the generic object in $\mathbf{RelCtx}$. We define this to be

$$[\![\vec{\alpha}, \vec{\beta} \mid x \colon \sigma(\vec{\alpha}), y \colon \sigma(\vec{\beta}) \mid \vec{R} \subset \vec{\alpha} \times \vec{\beta} \vdash \sigma[R](x, y)]\!]. \quad (6)$$

Given two vectors of types $\Xi \vdash \vec{\tau}, \vec{\tau}'$ we may reindex (6) in the composite fibration qr along $\langle \widehat{[\![\Xi \vdash \tau]\!]}, \widehat{[\![\Xi \vdash \tau']\!]} \rangle$ to get

$$[\![\Xi \mid x \colon \sigma(\vec{\tau}), y \colon \sigma(\vec{\tau}') \mid \vec{R} \subset \vec{\tau} \times \vec{\tau}' \vdash \sigma[R](x, y)]\!] \quad (7)$$

Suppose we are further given

$$\Xi \mid \Gamma \mid \Theta \vdash \rho_1 \subset \tau_1 \times \tau_1', \ldots, \rho_n \subset \tau_n \times \tau_n', \\ \Xi \mid \Gamma \vdash t \colon \sigma(\vec{\tau}/\vec{\alpha}) \quad \Xi \mid \Gamma \vdash s \colon \sigma(\vec{\tau}'/\vec{\alpha}).$$

Since definable relations correspond to propositions with two free variables, take ϕ_i such that $\rho_i = (x : \tau_i, y \colon \tau_i').\, \phi_i$. Now

$$\Psi^{-1}([\![\Xi \mid \Gamma, x \colon \tau, y \colon \tau' \mid \Theta \vdash \phi_i]\!]) \colon \\ [\![\Xi \mid \Gamma \mid \Theta]\!] \to [\![\Xi \mid R \subset \tau \times \tau']\!],$$

so we may reindex (7) along

$$\langle \langle I([\![s]\!]) \circ \pi, I([\![t]\!]) \circ \pi \rangle, \langle \Psi^{-1}([\![\psi_1]\!]), \ldots, \Psi^{-1}([\![\psi_n]\!]) \rangle \colon \\ [\![\Xi \mid \Gamma \mid \Theta]\!] \to [\![\Xi \mid x \colon \sigma(\vec{\tau}), y \colon \sigma(\vec{\tau}') \mid \vec{R} \subset \vec{\tau} \times \vec{\tau}']\!]$$

to get

$$[\![\Xi \mid \Gamma \mid \Theta \vdash \sigma[\vec{\rho}](s, t)]\!].$$

Theorem 3.7 (Soundness). *For any APL-structure, the interpretation defined above is sound with respect to the axioms and rules of Abadi and Plotkin's logic.*

Proof. Most of the proof is the same as 3.4. The axioms (1)-(3) are sound because J is a map of λ_2-fibrations. ☐

Theorem 3.8 (Completeness). *There exists an APL-structure with the property that any formula of Abadi and Plotkin's logic that holds in the APL-structure, i.e., is interpreted as true, may be proved in the logic.*

Proof. We will sketch how to construct this APL-structure out of the syntax of the logic. We will name the categories involved as in Definition 3.2. The category $\mathbf{Kind}$ has as objects natural numbers. A morphism from n to m is an m-vector of types with n free variables. These are identified up to renaming of variables.

The fiber of $\mathbf{Type}$ over n has as objects types with n free variables identified up to renaming of free variables. Morphisms are terms identified up to external equivalence. Reindexing in the fibrations $\mathbf{Type} \to \mathbf{Kind}$ is by substitution.

The fiber of $\mathbf{Ctx}$ over n has as objects contexts $\alpha_1, \ldots, \alpha_n \mid \Gamma \mid \Theta$ identified up to renaming of the α's. Morphisms in this category are vectors of terms and definable relations, where the terms are identified up to external equality and the definable relations are defined up to provable equivalence, i.e., a morphism into $\Xi \mid R \subset \sigma \times \tau$ from some context is a definable relation from σ to τ in that context, and two such definable relations ρ, ρ' are identified if $\forall x, y.\, \rho(x, y) \supset\subset \rho'(x, y)$ is provable in the logic.

Objects of $\mathbf{Prop}$ are propositions in the logic identified up to provable equivalence. These are ordered by implication in the logic.

The map J basically maps a type to its relational interpretation, and one can prove that this defines a functor using the Logical Relations Lemma proved in [11]. ☐

We refer to [9] for a more precise formulation of the above two theorems and their proofs.

4 Reasoning in Parametric APL-structures

An APL-structure contains a model of λ_2, which we may reason about using the internal language of the APL-structure. Thus we may formulate parametricity as a schema, exactly as in the original definition of relational parametricity [12]. For $\beta, \vec{\alpha} \vdash \sigma \colon \mathsf{Type}$ the parametricity schema is:

$$\forall \vec{\alpha} \colon \mathsf{Type}.\, \forall u \colon (\textstyle\prod \beta \colon \mathsf{Type}.\, \sigma(\beta, \vec{\alpha})). \\ u(\textstyle\prod \beta.\, \sigma)[eq_{\alpha_1}, \ldots, eq_{\alpha_n}]u. \quad (8)$$

For any type $\vec{\alpha} \vdash \sigma$, we may formulate the *identity extension schema* as

$$\forall \vec{\alpha} \colon \mathsf{Type}.\, \forall u, v \colon \sigma.\, (u(\sigma[eq_{\vec{\alpha}}])v \supset\subset u =_\sigma v). \tag{9}$$

It is easy to prove that the identity extension schema implies the parametricity schema. However, to prove the other direction one must use induction over the type structure of σ, which we cannot do since not all types in the model are neccessarily inductively defined.

Definition 4.1. A type $\vec{\alpha} \vdash \sigma$ with n free variables in a λ_2-fibration is *inductively defined* if it can be constructed from the α's and *closed* types using the type constructors of λ_2.

We may also define the extensionality schemas:

$$\forall t, u \colon \sigma \to \tau.\, (\forall x \colon \sigma.\, t\, x =_\tau u\, x) \supset t =_{\sigma \to \tau} u \tag{10}$$

and

$$\begin{gathered} \forall t, u \colon (\textstyle\prod \alpha \colon \mathsf{Type}.\, \sigma).\, (\forall \alpha \colon \mathsf{Type}.\, t\, \alpha =_\tau u\, \alpha) \supset \\ t =_{\prod \alpha \colon \mathsf{Type}.\tau} u. \end{gathered} \tag{11}$$

Recall [5] that a first order logic fibration has *very strong equality* if internal equality implies external equality. Examples of fibrations with very strong equality include subobject fibrations and regular subobject fibrations.

Definition 4.2. An APL-structure has *very strong equality* if each restriction of r to a fiber over $\mathbf{Kind}$ has so.

Definition 4.3. A *parametric APL-structure* is an APL-structure with very strong equality in which the identity extension schema (9) and the extensionality schemas (10) and (11) hold in the internal language of the model.

4.1 Consequences of parametricity

In this section we show that the definition of a parametric APL-structure allows one to prove the expected consequences of parametricity. We focus on initial algebras and final coalgebras.

Suppose we are given a type $\alpha \vdash \sigma \colon \mathsf{Type}$ with one free variable that is inductively constructed. Such a type induces a strong fibred functor (see [11])

$$\begin{tikzcd} \mathbf{Type} \arrow[rr, "\sigma"] \arrow[dr, "p"'] & & \mathbf{Type} \arrow[dl, "p"] \\ & \mathbf{Kind}. & \end{tikzcd}$$

More generally, we can consider the case where σ is *any* given polymorphically strong fibred functor, i.e., a fibred functor for which there exists a $t \colon \prod \alpha, \beta \colon \mathsf{Type}.\, (\alpha \to \beta) \to (\sigma(\alpha) \to \sigma(\beta))$ in the APL-structure such that $(t\alpha\beta)$ is a strength (see [6]) for σ.[2]

A family of (weak) initial algebras for σ is an indexed family of maps

$$(in_\Xi : \sigma(\tau_\Xi) \to \tau_\Xi)_{\Xi \in \mathbf{Kind}}$$

such that in each fiber $\mathbf{Type}_\Xi$, in_Ξ is a (weak) initial algebra for σ restricted to the fiber, and this family is closed under reindexing.

It is well known that if we define

$$\mu\alpha.\, \sigma[\alpha] = \textstyle\prod \alpha.\, ((\sigma(\alpha) \to \alpha) \to \alpha)$$

and $in : \sigma(\mu\alpha.\, \sigma[\alpha]) \to \mu\alpha.\, \sigma[\alpha]$ as

$$in\, z = \Lambda\alpha.\, \lambda f \colon \sigma[\alpha] \to \alpha.\, f(\sigma[\lambda x \colon \mu\alpha.\, \sigma[\alpha].\, x\, \alpha\, f]z)$$

then this defines a family of weak initial algebras. In Abadi and Plotkin's logic we can prove, using extensionality and the identity extension schema that any two algebra maps out of this weak initial algebra are internally equal [11, Theorem 8]. To conclude external equality we need very strong equality. Thus we have:

Theorem 4.4. *Suppose σ is a polymorphically strong fibred functor as above in a parametric APL-structure. The morphism in given above defines a family of initial algebras for σ.*

Likewise we may prove:

Theorem 4.5. *Suppose σ is a polymorphically strong fibred functor as above in a parametric APL-structure. Then σ has an indexed family of final coalgebras.*

5 Examples

In this section we show how a well known variant of the per-model (see, e.g., [5] and the references therein) can be seen as a concrete parametric APL-structure. The diagram of item 1 in the concrete model is:

$$\begin{tikzcd} & \mathbf{UFam}(\mathbf{RegSub}(\mathbf{Asm})) \arrow[d, "r"] \\ \mathbf{PFam}(\mathbf{Per}) \arrow[r, hook, "I"] \arrow[dr, "p"'] & \mathbf{UFam}(\mathbf{Asm}) \arrow[d, "q"] \\ & \mathbf{PPer} \end{tikzcd} \tag{12}$$

[2]The object part of a fibred functor is up to isomorphism given by an object in Type_Ω and thus we may use the internal language to define what it means to be polymorphically strong. See [9] for more details.

The fibration p is the fibration of [5, Def. 8.4.9]. We repeat the definition here. The category $\mathbf{Per}$ is the category of partial equivalence relations over $\mathbf{N}$ and the category $\mathbf{Asm}$ is the category of assemblies over $\mathbf{N}$. We use the notation $\mathbf{Per}_0$, $\mathbf{Asm}_0$ to denote the classes of objects. The category $\mathbf{PPer}$ is defined as

Objects Natural numbers.

Morphisms A morphism $f : n \to 1$ is a pair $(f^p, f^r)$where $f^p : \mathbf{Per}_0^n \to \mathbf{Per}_0$ is any map and

$$f^r \in \prod_{\vec{R},\vec{S}\in\mathbf{Per}^n} [\prod_{i\leq n} P(\mathbf{N}/R_i \times \mathbf{N}/S_i) \\ \to P(\mathbf{N}/f^p(\vec{R}) \times \mathbf{N}/f^p(\vec{S})]$$

is a map that satisfies $f^r(\overrightarrow{Eq}) = Eq$. We then define m to be the m-fold product of 1 by itself.

We can now define $\mathbf{PFam}(\mathbf{Per})$ as the indexed category with fiber over n defined as

Objects morphisms $n \to 1$ of $\mathbf{PPer}$.

Morphisms a morphism from f to g is an indexed family of maps $(\alpha_{\vec{R}})_{\vec{R}\in\mathbf{Per}^n}$ where

$$\alpha_{\vec{R}} : \mathbf{N}/f^p(\vec{R}) \to \mathbf{N}/g^p(\vec{R})$$

are tracked uniformly, i.e., there exists a code e such that for all $\vec{R}$ and $[n] \in \mathbf{N}/f^p(\vec{R})$, $\alpha_{\vec{R}}([n]) = [e \cdot n]$. Further, the morphism α should respect relations, that is, if $(a, b) \in f^r(\vec{A} \subset \mathbf{N}/\vec{R} \times \mathbf{N}/\vec{S})$ then $(\alpha_{\vec{R}}(a), \alpha_{\vec{S}}(b)) \in g^r(\vec{A})$.

In words, types of this model are pers and relations between pers. Reindexing is by composition.

The fiber category $\mathbf{UFam}(\mathbf{Asm})_n$ has as objects maps $\mathbf{Per}_0^n \to \mathbf{Asm}_0$ and as morphisms indexed families of maps in $\mathbf{Asm}$ that are uniformly tracked. The fibration r is just the regular subobject fibration, that is, an object over $(f(\vec{R}))_{\vec{R}}$ is a family of subsets $(A_{\vec{R}} \subseteq U(f(\vec{R})))_{\vec{R}}$, where $U : \mathbf{Asm} \to \mathbf{Set}$ is the forgetful functor. This fibration has very strong equality, as mentioned above.

We can define the reflexive graph since each type contains a relational interpretation (the map f^r). The APL-structure is parametric because the f^r satisfies the identity extension property. Thus we have:

Theorem 5.1. *The diagram in (12) defines a parametric APL-structure.*

Remark 5.2. In the construction above, we may replace the natural numbers with any partial combinatory algebra. We may generalize further to the case of relative realizability (see, e.g., [1]). This way we obtain a parametric APL-structure that is *not* well-pointed, c.f. the discussion in Section 1.

Example 5.3. Consider an arbitrary per $R \in \mathbf{Per}$. Since $\mathbf{Per} \cong \mathbf{PFam}(\mathbf{Per})_0$ (the fibre over the terminal object in $\mathbf{PPer}$), the fibred functor $F : \mathbf{PFam}(\mathbf{Per}) \to \mathbf{PFam}(\mathbf{Per})$ over $\mathbf{PPer}$ induced by the type α: Type $\vdash 1 + R \times \alpha$ gives an obvious functor $G : \mathbf{Per} \to \mathbf{Per}$ with $G(S) = 1 + R \times S$. By Theorem 4.4 applied to F, we then get an initial algebra for G in $\mathbf{Per}$. We may think of this initial algebra as consisting of R-lists.

6 Relation to Ma and Reynolds' Notion of Parametricity

In this section we compare the notion of parametricity defined above with Ma and Reynolds' notion of parametricity [7].

Definition 6.1. A λ_2 fibration $E \to B$ is parametric in the sense of Ma and Reynolds with respect to an indexed first order logic fibration $D \to E \to B$ if there exists a reflexive graph of λ_2-fibrations

$$\begin{pmatrix} E \\ \downarrow \\ B \end{pmatrix} \begin{array}{c} \longleftarrow \\ \longrightarrow \\ \longleftarrow \end{array} \begin{pmatrix} F \\ \downarrow \\ C \end{pmatrix}$$

such that the restriction of this graph to the fibres over the terminal objects becomes

$$E_1 \begin{array}{c} \longleftarrow \\ \longrightarrow \\ \longleftarrow \end{array} \mathbf{LR}(E_1) \tag{13}$$

where $\mathbf{LR}(E_1)$ is the category of relations on E_1 in the logic of D and maps preserving relations, and the three maps in the graph are the domain and codomain map, and the map mapping an object of $E_1$to its identity relation.

One difference between our notion of parametricity and Definition 6.1 is that in the latter it is not given what should be on the other side of the reflexive graph; in our definition it must be the fibration $\mathbf{Relations} \to \mathbf{RelCtx}$.

Another difference between the two is that in Definition 6.1 the reflexive graph (13) only says something about the closed types. To illustrate the difference, suppose $F \to C$ is in fact $\mathbf{Relations} \to \mathbf{RelCtx}$. Then the reflexive graph condition (13) tells us, for all closed types σ, that $J(\sigma) = eq_\sigma$, i.e.,

$$u\sigma v \iff u =_\sigma v.$$

Since the map in the reflexive graph commutes with reindexing, we have for all types $\vec{\alpha} \vdash \sigma\colon$ Type and $\vec{\tau}$ a vector of closed types

$$J(\llbracket\sigma(\vec{\tau})\rrbracket) = J(\llbracket\vec{\tau}\rrbracket^{*}\llbracket\vec{\alpha} \vdash \sigma\rrbracket) = \\ \llbracket eq_{\vec{\tau}}\rrbracket^{*} J(\llbracket\vec{\alpha} \vdash \sigma\rrbracket) = \llbracket\sigma(eq_{\vec{\tau}})\rrbracket.$$

In other words, the model satisfies a weak form of the Identity Extension Schema

$$\forall u, v\colon \sigma(\vec{\tau}).\, (u\sigma[eq_{\vec{\tau}}]v) \supset\subset u =_{\sigma(\vec{\tau})} v$$

where $\vec{\alpha} \vdash \sigma$ ranges over all types and $\vec{\tau}$ ranges over all *closed* types.

We have no proof that this weaker form of the identity extension schema is really weaker than the usual form, but if we try to replace the identity extension schema with the weaker form and repeat the proofs in Section 4, then the proof of Theorem 4.4 goes through, but the proof of Theorem 4.5 does not.

7 A Parametric Completion Process

In this section we give a description of a parametric completion process, that given a model of λ_2 internal to a quasi-topos satisfying certain requirements produces a parametric APL-structure. The construction is an extension of a parametric completion process given by Robinson and Rosolini [14] that produces models parametric in the sense of Definition 6.1 from internal models of λ_2. A consequence of our results is therefore that the models produced in [14] satisfy the consequences of parametricity (as expected but not shown in the literature before).

The concrete model of Section 5 is a result of the parametric completion process described in this section. We remark that there exist partial combinatory algebras for which the standard per model is *not* parametric [15]. It is still an open question whether there are partial combinatory algebras for which the *standard* per model is parametric. For the theory of internal models of λ_2 see [4] or [14]

Input for the parametric completion process is a quasi-topos E and a full internal model $\mathbf{D}$ of λ_2 in E, closed under products in E (i.e., the externalization of $\mathbf{D}$ into $E^{\rightarrow}$ preserves fibred products) and regular subobjects in E.

In a quasi-topos, the regular subobject fibration is a higher order logic fibration. This gives us:

Lemma 7.1. *The composable fibration* $Q \longrightarrow E^{\rightarrow} \xrightarrow{\mathrm{cod}} E$*, where* $Q \rightarrow E^{\rightarrow}$ *is the pullback*

$$\begin{tikzcd} Q \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{RegSub}_E \arrow[d] \\ E^{\rightarrow} \arrow[r, "\mathrm{dom}"] & E \end{tikzcd}$$

is an indexed first-order logic fibration with an indexed family of generic objects, simple products, simple coproducts and very strong equality.

We can form an internal fibration by using the Grothendieck construction on the functor $(d \in \mathbf{D}) \mapsto \Sigma^{d}$, where Σ is the regular subobject classifier. We think of this fibration as the internalisation of $\mathrm{RegSub}_E \rightarrow E$ restricted to $\mathbf{D}$ and write it as $\mathbf{Q} \rightarrow \mathbf{D}$. The fibration $\mathbf{Q} \rightarrow \mathbf{D}$ is a fibred ccc with simple products since its externalization is the restriction of $Q \rightarrow E^{\rightarrow}$ to $\mathrm{Fam}(\mathbf{D})$.

We define the category $\mathbf{LR}(\mathbf{D})$ to have as objects logical relations of $\mathbf{D}$ in the logic of $\mathbf{Q}$ and as morphisms pairs of morphisms in $\mathbf{D}$ that preserve relations. This is an internal cartesian closed category of E.

Assumption 1. The model $\mathbf{D}$ is "suitable for polymorphism", i.e., there exists Kan extensions of all functors $\mathbf{LR}(\mathbf{D})_0^n \rightarrow \mathbf{D}$ along projections.

Notice that this tells us in particular that $\mathbf{D}$ is closed under $\mathbf{LR}(\mathbf{D})_0$-indexed products.

Assumption 2. The logic $\mathbf{Q} \rightarrow \mathbf{D}$ is closed under $\mathbf{LR}(\mathbf{D})_0$-indexed products, i.e., if $(q_\rho \subset d_\rho)_{\rho \in \mathbf{LR}(\mathbf{D})_0}$ is a family of propositions in $\mathbf{Q} \rightarrow \mathbf{D}$ then

$$\{(x_\rho)_\rho\colon \textstyle\prod_\rho d_\rho \mid \forall \rho\colon \mathbf{LR}(\mathbf{D})_0.\, q_\rho(x_\rho)\}$$

is a proposition in the logic $\mathbf{Q} \rightarrow \mathbf{D}$.

7.1 The Completion Process

Let

$$G = \begin{tikzcd} \cdot \arrow[r, shift left=1.5] \arrow[r, shift right=1.5] & \cdot \arrow[l] \end{tikzcd}$$

denote the generic reflexive graph category, and consider the functor category E^{G}.

Now consider the functor $(\cdot)_0 : E^{G} \rightarrow E$ that maps a diagram $X_0 \rightleftarrows X_1$ to X_0, and consider (15) obtained from (14) by pullback along $(\cdot)_0$:

$$\begin{tikzcd} & Q \arrow[d] \\ \mathrm{Fam}(\mathbf{D}) \arrow[r, hook] \arrow[dr] & E^{\rightarrow} \arrow[d, "\mathrm{cod}"] \\ & E \end{tikzcd} \tag{14}$$

$$\begin{tikzcd} & P' \arrow[d] \\ T' \arrow[r, hook] \arrow[dr] & C' \arrow[d] \\ & E^{G}. \end{tikzcd} \tag{15}$$

If we extend (15) with the functor $\mathrm{Fam}\begin{pmatrix}\mathbf{LR}(\mathbf{D}) \\ \downarrow\uparrow\downarrow \\ \mathbf{D}\end{pmatrix} \to \mathrm{Fam}(\mathbf{D})$ defined by forgetting the $\mathbf{LR}(\mathbf{D})$ component we obtain (16):

$$
\begin{tikzcd}
 & & P' \arrow[d] \\
\mathrm{Fam}\begin{pmatrix}\mathbf{LR}(\mathbf{D})_0 \\ \downarrow\uparrow\downarrow \\ \mathbf{D}_0\end{pmatrix} \arrow[r] \arrow[drr] & T' \arrow[r, hook] \arrow[dr] & C' \arrow[d] \\
 & & E^G.
\end{tikzcd}
\tag{16}
$$

$$
\begin{tikzcd}
 & P \arrow[d] \\
T \arrow[r, "I"] \arrow[dr] & C \arrow[d] \\
 & K
\end{tikzcd}
\tag{17}
$$

Finally let K denote the full subcategory of E^G on powers of $\begin{pmatrix}\mathbf{LR}(\mathbf{D})_0 \\ \downarrow\uparrow\downarrow \\ \mathbf{D}_0\end{pmatrix}$. If we erase T' from (16) and pull the resulting diagram back along the inclusion of K into E^G we obtain (17). That is, T is the pullback of $\mathrm{Fam}\begin{pmatrix}\mathbf{LR}(\mathbf{D})_0 \\ \downarrow\uparrow\downarrow \\ \mathbf{D}_0\end{pmatrix}$.

Theorem 7.2. *The diagram (17) is a parametric APL-structure.*

Notice that the types in this model (the objects of T) are morphisms

$$\begin{pmatrix}\mathbf{LR}(\mathbf{D})_0 \\ \downarrow\uparrow\downarrow \\ \mathbf{D}_0\end{pmatrix}^n \to \begin{pmatrix}\mathbf{LR}(\mathbf{D})_0 \\ \downarrow\uparrow\downarrow \\ \mathbf{D}_0\end{pmatrix}$$

in E^G. So types contain both the usual interpretation (the map $f_0 : \mathbf{D}_0^n \to \mathbf{D}_0$) and a relational interpretation (the map $f_1 : \mathbf{LR}(\mathbf{D})_0^n \to \mathbf{LR}(\mathbf{D})_0$). But since the map $\mathrm{Fam}\begin{pmatrix}\mathbf{LR}(\mathbf{D})_0 \\ \downarrow\uparrow\downarrow \\ \mathbf{D}_0\end{pmatrix} \to T'$ forgets the relational interpretation, the logic on types, given by P, is given only by the logic on the usual interpretation of the types. Thus a logical relation in this model between types f and g is a relation in the sense of the logic Q between $\coprod_{b\in\mathbf{D}_0^n} f_0^{-1}(b) \to \mathbf{D}_0^n$ and $\coprod_{b\in\mathbf{D}_0^n} g_0^{-1}(b) \to \mathbf{D}_0^n$.

Notice also that the relational interpretation of a type (given by f_1) is in a sense parametric since the diagram

$$
\begin{tikzcd}
\mathbf{LR}(\mathbf{D})_0^n \arrow[r, "f_1"] & \mathbf{LR}(\mathbf{D})_0 \\
\mathbf{D}_0^n \arrow[r, "f_0"] \arrow[u, "i"] & \mathbf{D}_0 \arrow[u, "i"']
\end{tikzcd}
$$

is required to commute. This is basically the reason why this model is parametric.

Consider a morphism ξ between types f and g in the model. At first sight, such a morphism is a pair of morphism (ξ_0, ξ_1) with $\xi_i : f_i \to g_i$. But morphisms in $\mathbf{LR}(\mathbf{D})$ are given by pairs of maps in $\mathbf{D}$, and commutativity of

$$
\begin{tikzcd}
\mathbf{LR}(\mathbf{D})_0^n \arrow[r, "\xi_1"] \arrow[d, "\partial_i"'] & \mathbf{LR}(\mathbf{D})_1 \arrow[d, "\partial_i"] \\
\mathbf{D}_0^n \arrow[r, "\xi_0"] & \mathbf{D}_1
\end{tikzcd}
$$

tells us that ξ_1 must be given by (ξ_0, ξ_0). Thus *morphisms between types are morphisms between the usual interpretations of types preserving the relational interpretations.*

We do not have room for a proof of Theorem 7.2 here (see [9] for a detailed proof), but we will give a sketch of the proof. It is well known [14] that $T \to K$ is a λ_2-fibration, and since Lemma 7.1 gives most of what is needed to prove Theorem 7.2, we will just sketch why the model has relational interpretations of all types.

Consider the graph $\mathbf{W}$:

$$
\begin{tikzcd}
\mathbf{LR}(\mathbf{D}) \arrow[d, shift left=3] \arrow[d] \arrow[d, shift right=3] & \mathbf{LR}(\mathbf{D}) \arrow[dl] \arrow[dr] & \mathbf{LR}(\mathbf{D}) \arrow[d, shift left=3] \arrow[d, shift right=3] \\
\mathbf{D} \arrow[u] & & \mathbf{D} \arrow[u]
\end{tikzcd}
$$

which is an internal category in the obvious graph category over E. As in [14] there is a reflexive graph of λ_2-fibrations

$$
\begin{pmatrix} T \\ \downarrow \\ K \end{pmatrix}
\begin{array}{c}\longleftarrow \\ \longrightarrow \\ \longleftarrow\end{array}
\begin{pmatrix} \mathrm{Fam}(\mathbf{W}) \\ \downarrow \\ \{\mathbf{W}^n \mid n \in \mathbf{N}\} \end{pmatrix}.
\tag{18}
$$

However, the definition of a parametric APL-structure requires a reflexive graph with $\mathbf{Relations} \to \mathbf{RelCtx}$ (constructed from (17) as in Lemma 3.5) on the right-hand side. Such a graph exists because the fibration on the right in (18) is a subfibration of $\mathbf{Relations} \to \mathbf{RelCtx}$. To see that, notice that objects of $\mathrm{Fam}(\mathbf{W})$ are pairs of maps f, g between types and a map $\phi : \mathbf{LR}(\mathbf{D})_0^n \to \mathbf{LR}(\mathbf{D})_0$ such that

$$
\begin{tikzcd}
 & \mathbf{LR}(\mathbf{D})_0^n \arrow[rr, "\phi"] \arrow[ddl] \arrow[ddr] & & \mathbf{LR}(\mathbf{D})_0 \arrow[dl] \arrow[dr] & \\
 & & \mathbf{D}_0 & & \mathbf{D}_0 \\
\mathbf{D}_0^n \arrow[urr, "f_0"] & & \mathbf{D}_0^n \arrow[urr, "g_0"] & &
\end{tikzcd}
\tag{19}
$$

commutes. Since $\mathbf{LR}(\mathbf{D})_0 \cong \coprod_{\alpha,\beta\in\mathbf{D}} \Sigma^{\alpha\times\beta}$ such maps correspond to relations

$$[\![\vec{\alpha}, \vec{\beta} \mid \vec{R} \subset \vec{\alpha} \times \vec{\beta} \vdash \phi \subset f(\vec{\alpha}) \times g(\vec{\beta})]\!].$$

8 Conclusions and future work

We have defined the notion of a parametric APL-structure and proved that it provides sound and complete models for Abadi and Plotkin's logic for parametricity. For parametric APL-structures we can prove the expected consequences of parametricity using the internal logic. These consequences have to our knowledge not been proved in general for models parametric in the sense of Ma & Reynolds.

We have presented a family of parametric models, and proved that a slight extension of the parametric completion process of [14] produces parametric APL-structures.

In subsequent papers, we will show how to modify the parametric completion process to produce domain-theoretic parametric models and how to extend the notion of APL-structure to include models of linear/non-linear λ_2 [10].

Acknowledgements

We are grateful to Pino Rosolini for insightful discussions. The research in this paper was partially supported by the Danish Technical Research Council (Grant No.: 56–00–0309).

A Composable Fibrations

This appendix is concerned with composable fibrations. It contains definitions used in the text.

Suppose we are given two composable fibrations $F \xrightarrow{p} E \xrightarrow{q} B$. It is easily seen that the composite qp is a fibration, and if both fibrations are cloven with chosen lifts denoted $u \mapsto \bar{u}$ then $u \mapsto \bar{\bar{u}}$ is a cleavage for the composite fibration. If p, q are split, so is qp.

Definition A.1. We say that $(\Omega_A)_{A \in \mathrm{Obj}B}$ is an indexed family of generic objects for the composable pair of fibrations (p, q) if for all A, $\Omega_A \in \mathrm{Obj}E_A$ is a generic object for the restriction of p to E_A and if the family is closed under reindexing, ie. for all morphisms $u : A \to B$ in B $u^*(\Omega_B) = \Omega_A$.

Definition A.2. We say that (p, q) has indexed (simple) products /coproducts /equality if each restriction of p to a fiber of q has the same satisfying the Beck-Chevalley condition, and these commute with reindexing, ie. if u is a map in B then there is a natural isomorphism $\bar{u}^*\prod_f \cong \prod_{u^*f}\bar{u}^*$ or $\bar{u}^*\coprod_f \cong \coprod_{u^*f}\bar{u}^*$ (this can also be viewed as a Beck-Chevalley condition).

Definition A.3. We say that (p, q) is an *indexed first order logic fibration* if p is a bicartesian closed preorder fibration and (p, q) has indexed simple products, indexed simple coproducts and indexed equality.

Definition A.4. We say that the composable fibration (p, q) has (co-)products with respect to a map u in B if p has (co-)products with respect to any lift $\bar{u}$ of u satisfying the Beck-Chevalley condition.

References

[1] L. Birkedal. Developing theories of types and computability via realizability. *Electronic Notes in Theoretical Computer Science*, 34:viii+282, 2000. Book version of PhD-thesis.

[2] B. Dunphy. *Parametricity as a notion of uniformity in reflexive graphs*. PhD thesis, University of Illinois, 2002.

[3] R. Hasegawa. Categorical data types in parametric polymorphism. *Mathematical Structures in Computer Science*, 4:71–109, 1994.

[4] J. Hyland, E. Robinson, and G. Rosolini. The discrete objects in the effective topos. *Proc. London Math. Soc.*, 3(60):1–36, 1990.

[5] B. Jacobs. *Categorical Logic and Type Theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science Publishers B.V., 1999.

[6] A. Kock. Strong functors and monoidal monads. *Archiv der Mathematik*, XXIII:113–120, 1972.

[7] Q. Ma and J. Reynolds. Types, abstraction, and parametric polymorphism, part 2. In S. Brookes, M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *Mathematical Foundations of Programming Semantics*, volume 598 of *Lecture Notes in Computer Science*, pages 1–40. Springer-Verlag, 1992.

[8] J. Mitchell. *Foundations for Programming Languages*. MIT Press, 1996.

[9] R. Møgelberg and L. Birkedal. On the definition of parametricity. Technical report, The IT University of Copenhagen, Copenhagen, Denmark, October 2003. Available at `http://www.itu.dk/people/mogel/paramdef.ps`.

[10] G. Plotkin. Second order type theory and recursion. Notes for a talk at the Scott Fest, Feb. 1993.

[11] G. Plotkin and M. Abadi. A logic for parametric polymorphism. In *Typed lambda calculi and applications (Utrecht, 1993)*, volume 664 of *Lecture Notes in Comput. Sci.*, pages 361–375. Springer, Berlin, 1993.

[12] J. Reynolds. Types, abstraction, and parametric polymorphism. *Information Processing*, 83:513–523, 1983.

[13] J. Reynolds. Polymorphism is not set-theoretic. In G. Kahn, D. B. MacQueen, and G. D. Plotkin, editors, *Semantics of Data Types*, volume 173 of *Lecture Notes in Computer Science*, pages 145–156. Springer-Verlag, 1984.

[14] E. Robinson and G. Rosolini. Reflexive graphs and parametric polymorphism. In S. Abramsky, editor, *Proc. 9th Symposium in Logic in Computer Science*, pages 364–371, Paris, 1994. I.E.E.E. Computer Society.

[15] I. Rummelhoff. Polynat in per-models. Manuscript. University of Oslo. Available from http://www.math.uio.no/˜ivarru/research/papers.html, 2003.

[16] R. Seely. Categorical semantics of higher-order polymorphic lambda calculus. *The Journal of Symbolic Logic*, 52(4):969–989, Dec. 1987.

[17] P. Wadler. Theorems for free! In *4'th Symposium on Functional Programming Languages and Computer Architecture, ACM, London*, Sept. 1989.