Theory and Applications of Categories, Vol. 20, No. 7, 2008, pp. 116–151.

CATEGORY-THEORETIC MODELS OF LINEAR ABADI & PLOTKIN LOGIC

LARS BIRKEDAL AND RASMUS E. MØGELBERG AND RASMUS L. PETERSEN

Abstract. This paper presents a sound and complete category-theoretic notion of models for Linear Abadi & Plotkin Logic [Birkedal et al., 2006], a logic suitable for reasoning about parametricity in combination with recursion. A subclass of these called *parametric LAPL structures* can be seen as an axiomatization of domain theoretic models of parametric polymorphism, and we show how to solve general (nested) recursive domain equations in these. Parametric LAPL structures constitute a general notion of model of parametricity in a setting with recursion. In future papers we will demonstrate this by showing how many different models of parametricity and recursion give rise to parametric LAPL structures, including Simpson and Rosolini's set theoretic models [Rosolini and Simpson, 2004], a syntactic model based on Lily [Pitts, 2000, Bierman et al., 2000] and a model based on admissible pers over a reflexive domain [Birkedal et al., 2007].

1. Introduction

When Reynolds introduced relational parametricity in 1983 [Reynolds, 1983] he argued that it could be used to prove representation independence results for abstract data types. Since then a large number of applications of the abstraction property of parametricity have been suggested, for example in security [Tse and Zdancewic, 2004] and to model local variables [O'Hearn and Tennent, 1995]. From a type theoretic perspective, relational parametricity is interesting because it allows rich type systems to be constructed from a few basic type constructors. For example in the second order lambda calculus, general inductive and coinductive types can be constructed using simply polymorphism and function space type constructors. For real programming, one is of course not just interested in a strongly terminating calculus such as the second-order lambda calculus, but also in a language with full recursion. Thus in [Reynolds, 1983] Reynolds also asked for a parametric *domain-theoretic* model of polymorphism.

However, relational parametricity is too strong a principle to be simply combined with recursion in the second order lambda calculus. One way to see this is that parametricity gives encodings of coproducts, and it is well known that the combination of coproducts, products, exponentials and fixed points exist only in the trivial case of all types being

This work was partly supported by the Danish Agency for Science Technology and Innovation
Received by the editors 2006-04-20 and, in revised form, 2008-04-09.
Transmitted by Valeria de Paiva. Published on 2008-04-11.
2000 Mathematics Subject Classification: 03B70, 18C50, 68Q55.
Key words and phrases: Parametric polymorphism, categorical semantics, axiomatic domain theory, recursive types, fibrations.

isomorphic [Huwig and Poigné, 1990]. Inspired by domain theory Plotkin suggested an elegant solution to this problem [Plotkin, 1993a, Plotkin, 1993b], which was to study a dual intuitionistic / linear lambda calculus with polymorphism and fixed points. Parametricity should then be available only in the linear part of the calculus and give encodings of datatypes in this, including initial algebras and final coalgebras. Moreover, by Freyd's theory of algebraically compact categories [Freyd, 1990a, Freyd, 1990b, Freyd, 1991] the inductive and coinductive types in combination with fixed points would give solutions to recursive type equations in the linear calculus.

Recently, Pitts and coworkers [Pitts, 2000, Bierman et al., 2000] have presented the language Lily, essentially Plotkin's polymorphic intuitionistic / linear lambda calculus equipped with an operational semantics, and shown how Plotkin's encodings can be used in the syntactic setting.

In parallel with the work presented here, Rosolini and Simpson [Rosolini and Simpson, 2004] have shown how to construct parametric domain-theoretic models in intuitionistic set-theory using synthetic domain-theory. Moreover, they have shown how to give a computationally adequate denotational semantics of Lily.

This paper presents a unifying category-theoretic framework for parametric models of Polymorphic Intuitionistic / Linear Lambda calculus with a fixed point combinator Y (a calculus henceforth referred to as PILL_Y). The basis is Linear Abadi & Plotkin Logic (LAPL), an adaptation of Abadi & Plotkin's logic for parametricity on the second order lambda calculus [Plotkin and Abadi, 1993] to PILL_Y, presented by the authors in [Birkedal et al., 2006]. In LAPL the parametricity principle for PILL_Y can be formulated and the above mentioned consequences of parametricity including the solutions to recursive type equations can be proved valid. In this paper we present the notion of LAPL structure which is a sound and complete category-theoretic notion of model of LAPL, and we define a subclass of these called *parametric* LAPL structures to be LAPL structures in which the parametricity principle holds and which satisfy a technical condition called very strong equality. Using reasoning in LAPL we show how to solve recursive domain equations in parametric LAPL structures, and moreover do so in full generality considering parametrized recursive domain equations (needed to model nested recursive types). The recursive types satisfy a parametrised version of the initial dialgebra property, a universal property generalising initial algebras and final coalgebras to general recursive types. We formulate and prove this property using fibred category theory. These results show that the notion of parametric LAPL structure is indeed a useful notion of parametric model of PILL_Y, since these satisfy semantic versions of the consequences of parametricity known from logic.

Forthcoming papers will show how the Lily language and the set theoretic constructions mentioned above can be seen as constructing LAPL structures. In [Birkedal et al., 2007] we present a parametric LAPL structure based on admissible pers over a reflexive domain. This model was first suggested by Plotkin [Plotkin, 1993a], but the details have to our knowledge never been worked out before. In [Møgelberg, 2005b] the second author shows how the parametric completion process as in [Robinson and Rosolini, 1994] can be adapted to construct a parametric LAPL structures from a model of PILL_Y. The examples show that the notion of parametric LAPL structure is general enough to cover many different models, and the general results about parametric LAPL structures apply to all these models, giving rigourous proofs of consequences of parametricity.

The work presented here builds upon previous work by the first two authors on categorical models of Abadi-Plotkin's logic for parametricity [Birkedal and Møgelberg, 2005], which includes detailed proofs of consequences of parametricity for polymorphic lambda calculus and also includes a description of a parametric completion process that given an internal model of polymorphic lambda calculus produces a parametric model. It is not necessary to be familiar with the details of [Birkedal and Møgelberg, 2005] to read the present paper, although Appendix A of [Birkedal and Møgelberg, 2005], contains some definitions and theory concerning composable fibrations which are only briefly summarised in this paper.

1.1. OUTLINE. The remainder of this paper is organized as follows. Section 2 briefly recalls PILL_Y and Linear Abadi & Plotkin Logic (LAPL), and the main results of [Birkedal et al., 2006], and Section 3 recalls the definition of PILL_Y models. Building upon the notion of PILL_Y model Section 4 defines the notion of LAPL structure and proves soundness and completeness of the interpretation of LAPL in such structures. In Section 5 we present our definition of a parametric LAPL structure and prove that one may solve recursive domains equations in such.

2. Linear Abadi-Plotkin Logic

This section briefly recalls Linear Abadi-Plotkin Logic (LAPL) as defined in [Birkedal et al., 2006]. LAPL is a logic for reasoning about parametricity for Polymorphic Intuitionistic Linear Lambda calculus with fixed points (PILL_Y). The logic is based on Abadi and Plotkin's logic for parametricity [Plotkin and Abadi, 1993] for the second-order lambda calculus.

2.1. PILL_Y. The calculus PILL_Y is essentially Barber and Plotkin's DILL [Barber, 1997] extended with polymorphism and a fixed point combinator. Types are formed using the grammar

$$\sigma ::= \alpha \mid I \mid \sigma \otimes \tau \mid \sigma \multimap \tau \mid !\sigma \mid \textstyle\prod \alpha.\, \sigma.$$

where α ranges over type variables. PILL_Y can be seen as a type theory for domain theory and polymorphism, and even though polymorphism cannot be modelled in ordinary domain theory it is useful to keep the domain theoretic intuition in mind: types are domains (complete partial orders with a least element), $\otimes$ denotes the smash product, $\multimap$ the domain of strict continuous maps and ! denotes lifting. One can make this intuition precise by constructing a DILL model based on the category of domains and strict continuous maps, but we will not do that here. One way to extend the domain theoretic intuition to polymorphism is by considering partial equivalence relations. This is used to construct a PILL_Y model and a parametric LAPL structure in [Birkedal et al., 2007].

Terms of PILL_Y are written with explicit contexts as

$$\alpha_1, \dots, \alpha_k \mid x_1 \colon \sigma_1, \dots, x_n \colon \sigma_n; x'_1 \colon \sigma'_1, \dots, x'_m \colon \sigma'_m \vdash t \colon \tau,$$

where the list $\alpha_1, \dots, \alpha_k$ — called the kind context — contains all the type variables occurring freely in the types σ_i, σ'_i and τ. The term t has two contexts of free term variables: the list of x's is called the intuitionistic type context and is often denoted Γ, and the list of x''s is called the linear type context, often denoted Δ. Variables in the linear context can occur linearly in terms and variables in the intuitionistic context can occur intuitionistically. We use the horizontal line $-$ to denote an empy context, in particular $- \mid -; - \vdash t \colon \sigma$ denotes a closed term of closed type.

The grammar for terms is:

$$\begin{aligned} t ::= & x \mid \star \mid Y \mid \lambda^\circ x \colon \sigma.t \mid t\, t \mid t \otimes t \mid !t \mid \Lambda\alpha \colon \mathsf{Type}.\, t \mid t(\sigma) \mid \\ & \text{let } x \colon \sigma \otimes y \colon \tau \text{ be } t \text{ in } t \mid \text{let } !x \colon \sigma \text{ be } t \text{ in } t \mid \text{let } \star \text{ be } t \text{ in } t \end{aligned}$$

The construction $\lambda^\circ x \colon \sigma.t$ abstracts *linear* variables in terms and constructs terms of type $\sigma \multimap \tau$, which we think of as a linear function space. Intuitionistic function space can be defined using the Girard encoding $\sigma \to \tau = !\sigma \multimap \tau$, and there is a corresponding lambda abstraction abstracting intuitionistic variables. Because of the encoding of $\to$ application of an intuitionistic function $f \colon \sigma \to \tau$ to an input $x \colon \sigma$ becomes $f(!x)$. Using the Girard encodings, the type of the fixed point combinator Y can be written as $\prod \alpha.\, (\alpha \to \alpha) \to \alpha$. The subset of PILL_Y without the fixed point combinator is referred to as PILL.

Terms of PILL_Y are identified up to an equality relation, which we shall call *external equality* (as opposed to provable equality in the logic LAPL). Rules for external equality includes β and η rules, but we refer to [Birkedal et al., 2006] for a definition.

In the domain theoretic intuition, linearity in PILL_Y is modelled by strictness, i.e., we think of a term

$$- \mid x_1 \colon \sigma_1, \dots, x_n \colon \sigma_n; x'_1 \colon \sigma'_1, \dots, x'_m \colon \sigma'_m \vdash t \colon \tau$$

as a continuous function from $\prod_{i \le n} \sigma_i \times \prod_{j \le m} \sigma'_j$ to τ which maps a tuple

$$(x_1, \dots, x_n, x'_1; \dots, x'_m)$$

to the least element $\bot$ if any one of the x'_j is $\bot$.

2.2. LAPL. As mentioned, Linear Abadi-Plotkin Logic is a second order logic suitable for reasoning about parametricity over PILL_Y. As in Abadi & Plotkin's logic for parametricity on the second order lambda calculus [Plotkin and Abadi, 1993, Birkedal and Møgelberg, 2005], the logic contains relations and the relational interpretation of types as basic building blocks. We mention two peculiarities about the logic. First, unlike the general PILL_Y terms propositions in the logic have no linear context of term variables, only an intuitionistic one. This means that even though PILL_Y is dual linear / intuitionistic the reasoning in the logic is purely intuitionistic. We may still reason about linearity of terms of PILL_Y since we may consider terms of linear function space type. This restriction of the logic has been chosen for simplicity and because the reasoning that we need in the logic to prove properties of polymorphic encodings in PILL_Y is purely intuitionistic.

The second peculiarity is that the logic has two classes of relations, a class of ordinary relations and a class of "admissible" relations. In the domain theoretic intuition, a relation is just a subset of the product of two domains, whereas an admissible relation must relate $\perp$ to $\perp$ and be closed under least upper bounds of chains. Indeed the main example of an admissible relation in LAPL is the graph of a linear function $f\colon \sigma \multimap \tau$, and the main example of a relation that is not in general admissible is the graph of an intuitionistic function $f\colon \sigma \to \tau$. The introduction of admissible relations in the logic is necessary because, as noticed by Plotkin, adding an unrestricted parametricity principle to PILL_Y gives an inconsistent logic, a problem which can be remedied by restricting the formulation of the parametricity principle to the class of admissible relations.

The logic has three types of judgements: one for propositions, one for relations and one for admissible relations:

$$\Xi \mid \Gamma \mid \Theta \vdash \phi\colon \mathsf{Prop}$$
$$\Xi \mid \Gamma \mid \Theta \vdash \rho\colon \mathsf{Rel}(\tau, \tau')$$
$$\Xi \mid \Gamma \mid \Theta \vdash \rho\colon \mathsf{AdmRel}(\tau, \tau').$$

Here, Ξ and Γ are respectively the kind context and the intuitionistic type context as in PILL_Y, and Θ is a context of relational variables of the form

$$R_1\colon \mathsf{Rel}(\tau_1, \tau_1'), \dots, R_n\colon \mathsf{Rel}(\tau_n, \tau_n'), S_1\colon \mathsf{AdmRel}(\omega_1, \omega_1'), \dots, S_m\colon \mathsf{AdmRel}(\omega_m, \omega_m').$$

The grammar for propositions is

$$\begin{aligned} \phi \quad ::= \quad & (t =_\sigma u) \mid \rho(t, u) \mid \ \phi \supset \psi \mid \perp \mid \top \mid \phi \wedge \psi \mid \phi \vee \psi \mid \forall \alpha\colon \mathsf{Type}.\, \phi \mid \\ & \forall x\colon \sigma.\, \phi \mid \forall R\colon \mathsf{Rel}(\sigma, \tau).\, \phi \mid \forall S\colon \mathsf{AdmRel}(\sigma, \tau).\, \phi \mid \\ & \exists \alpha\colon \mathsf{Type}.\, \phi \mid \exists x\colon \sigma.\, \phi \mid \exists R\colon \mathsf{Rel}(\sigma, \tau).\, \phi \mid \exists S\colon \mathsf{AdmRel}(\sigma, \tau).\, \phi \end{aligned}$$

where ρ is a relation (admissible or not). As an example of a formation rule we mention the one for equality

$$\frac{\Xi \mid \Gamma; - \vdash t\colon \sigma \qquad \Xi \mid \Gamma; - \vdash u\colon \tau}{\Xi \mid \Gamma \mid \Theta \vdash t =_\sigma u\colon \mathsf{Prop}}$$

where the judgements in the hypothesis are typing judgements in PILL_Y.

The grammar for relations is

$$\rho ::= R \mid (x\colon \sigma, y\colon \tau).\, \phi \mid \sigma[\vec{\rho}]$$

Relations can be formed from proposition and vice versa, as determined by the two rules

$$\frac{\Xi \mid \Gamma \ \mid \Theta \vdash \rho\colon \mathsf{Rel}(\sigma, \tau) \qquad \Xi \mid \Gamma; - \vdash t\colon \sigma \qquad \Xi \mid \Gamma; - \vdash u\colon \tau}{\Xi \mid \Gamma \mid \Theta \vdash \rho(t, u)\colon \mathsf{Prop}}$$

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

The admissible relations are a subset of the relations closed under a collection of rules for which we refer to [Birkedal et al., 2006] as we do for all other details in this section. Examples of relations include equality relations eq_τ defined as $(x\colon \tau, y\colon \tau).\, x =_\tau y$, graphs of linear maps $\langle f\rangle\colon \mathsf{Rel}(\sigma,\tau)$ for $f\colon \sigma \multimap \tau$ defined as $(x\colon \sigma, y\colon \tau).\, y =_\tau f(x)$ and graphs of intuitionistic maps $\langle g\rangle\colon \mathsf{Rel}(\sigma,\tau)$ for $g\colon \sigma \to \tau$ defined as $(x\colon \sigma, y\colon \tau).\, y =_\tau f(!x)$ of which the first two are admissible, but the last generally is not.

We introduce some notation that is needed later in the text: if $\rho\colon \mathsf{Rel}(\sigma,\tau)$ and $f\colon \sigma' \multimap \sigma, g\colon \tau' \multimap \tau$, we write $(f,g)^*\rho$ for

$$(x\colon \sigma', y\colon \tau').\, \rho(f(x), g(y))$$

The relational interpretation of types is the rule

$$\frac{\alpha_1,\dots,\alpha_n \vdash \sigma(\vec{\alpha})\colon \mathsf{Type} \qquad \Xi \mid \Gamma \mid \Theta \vdash \rho_1\colon \mathsf{AdmRel}(\tau_1,\tau_1'),\dots,\rho_n\colon \mathsf{AdmRel}(\tau_n,\tau_n')}{\Xi \mid \Gamma \mid \Theta \vdash \sigma[\vec{\rho}]\colon \mathsf{AdmRel}(\sigma(\vec{\tau}),\sigma(\vec{\tau}'))} \tag{1}$$

The PILL_Y type constructors $\multimap, \otimes, !, \prod\alpha.\,(-)$ may all be given relational interpretations, by for example defining $\rho \multimap \rho'$ for $\rho\colon \mathsf{Rel}(\sigma,\tau)$, $\rho'\colon \mathsf{Rel}(\sigma',\tau')$ to be

$$(f\colon \sigma \multimap \sigma', g\colon \tau \multimap \tau').\, \forall x\colon \sigma, y\colon \tau.\, \rho(x,y) \supset \rho'(f(x), g(y)).$$

Using these, the relational interpretation of types in *pure* PILL_Y can be defined by structural induction. However, the relational interpretation of types in LAPL is stronger. This is because LAPL can be used for reasoning about not just pure PILL_Y but also other PILL_Y-calculi with added type or term constants, the most prominent example being the internal language of a given PILL_Y model. Such models may contain open type constants such as $\alpha \vdash ?(-)\colon \mathsf{Type}$, and (1) says that each of these need to have a relational interpretation. The inductive definition sketched above is then captured in axioms for LAPL stating for example that $(\sigma \multimap \tau)[R] \equiv \sigma[R] \multimap \tau[R]$.

Implication in the logic is formally written as $\Xi \mid \Gamma \mid \Theta \mid \phi_1,\dots,\phi_n \vdash \psi$, meaning that in the context $\Xi \mid \Gamma \mid \Theta$ the propositions $\phi_1,\dots,\phi_n$ collectively imply ψ. For $\rho,\rho'\colon \mathsf{Rel}(\sigma,\tau)$ we use shorthand $\Xi \mid \Gamma \mid \Theta \mid \top \vdash \rho \equiv \rho'$ for

$$\Xi \mid \Gamma \mid \Theta \mid \top \vdash \forall x\colon \sigma, y\colon \tau.\, \rho(x,y) \supset\subset \rho'(x,y).$$

Using this notation we have already informally stated one axiom of the logic, but we refer to [Birkedal et al., 2006] for details.

In LAPL we can formulate the identity extension schema $\sigma[\vec{eq}_\tau] \equiv eq_{\sigma(\vec{\tau})}$. Identity extension implies the parametricity schema

$$\forall x\colon \textstyle\prod\alpha.\,\sigma.\, \forall\alpha,\beta.\, \forall R\colon \mathsf{AdmRel}(\alpha,\beta).\, \sigma[R](x\,\alpha, x\beta).$$

Instantiating the parametricity schema with the type of the fixed point combinator

$$Y\colon \textstyle\prod\alpha.\, (\alpha \to \alpha) \to \alpha$$

$$
\begin{array}{rcl}
\sigma & \cong & \prod\alpha.\,(\sigma \multimap \alpha) \multimap \alpha \\
\sigma \otimes \tau & \cong & \prod\alpha.\,(\sigma \multimap \tau \multimap \alpha) \multimap \alpha \\
I & \cong & \prod\alpha.\,\alpha \multimap \alpha \\
0 & = & \prod\alpha.\,\alpha \\
1 & = & \prod\alpha.\,\alpha \\
\sigma + \tau & = & \prod\alpha.\,(\sigma \multimap \alpha) \to (\tau \multimap \alpha) \to \alpha \\
\sigma \times \tau & = & \prod\alpha.\,(\sigma \multimap \alpha) + (\tau \multimap \alpha) \multimap \alpha \\
\mathbb{N} & = & \prod\alpha.\,(\alpha \multimap \alpha) \to \alpha \multimap \alpha \\
\coprod\alpha.\,\sigma & = & \prod\beta.\,(\prod\alpha.\,\sigma \multimap \beta) \multimap \beta \\
\mu\alpha.\,\sigma & = & \prod\alpha.\,(\sigma \multimap \alpha) \to \alpha \\
\nu\alpha.\,\sigma & = & \coprod\alpha.\,!(\alpha \multimap \sigma) \otimes \alpha
\end{array}
$$

Figure 1: Types definable using parametricity

and applying it to the graph of a linear function $h\colon \sigma \multimap \tau$ we find Plotkin's principle: if $f\colon \sigma \to \sigma$ and $g\colon \tau \to \tau$ are such that $h \circ f = g \circ h$ then $h(Y(!f)) = Y(!g)$. The restriction to admissible relations of the universal quantification in the parametricity principle restricts us from instantiating it with the graphs of intuitionistic functions, just like Plotkin's principle in domain theory only holds for strict maps h.

In [Birkedal et al., 2006] we show how the identity extension schema implies correctness of encodings of certain datatypes in PILL_Y, some of which are listed in Figure 1. What is meant by correctness of these is that it is provable in the logic that these satisfy the usual universal properties with respect to *linear* functions. For example, in the last two of the type encodings in Figure 1, the type variable α is assumed to occur only positively in σ, and such types induce endofunctors on the category of PILL_Y types with linear terms as morphisms, for which the types defined are the inductive and coinductive datatypes respectively. One might be tempted to try to define initial algebras and final coalgebras for the functors induced by types on the category of PILL_Y types and intuitionistic terms using the encodings from second order lambda calculus. Such an attempt will fail because the parametricity arguments from second order lambda calculus involve instantiations of the parametricity principle with graphs of general intuitionistic functions, and so can not be carried out in LAPL.

The full type theoretical strength of the combination of parametricity and recursion is however, that we can find solutions to general recursive type equations in PILL_Y. More precisely, using Freyd's theory of algebraically compact categories [Freyd, 1990a, Freyd, 1990b, Freyd, 1991] the combination of initial algebras, final coalgebras and fixed points allows us to construct, for *any* type expression $\alpha \vdash \sigma$ in pure PILL_Y, a type τ such that $\sigma(\tau) \cong \tau$. The constructed solutions moreover satisfy the initial dialgebra property, a universal property generalising initial algebras and final coalgebras.

Further details can be found in [Birkedal et al., 2006]. In Section 5.8 we explain in detail the semantic consequences of this construction, defining a semantic notion of recursive domain equations that can be solved in parametric LAPL structures and formulating the universal properties satisfied by the constructed solutions.

As mentioned, the universal conditions satisfied by the polymorphic type encodings, in particular the initial dialgebra property of the recursive types, are formulated with respect to linear maps in PILL_Y rather than intuitionistic maps. This corresponds with the domain theoretic intuition for PILL_Y given earlier, as the category $\mathbf{Cppo}_\perp$ of domains and strict continuous maps is algebraically compact and the category **Cppo** of domains and continuous maps is not.

3. PILL_Y models

In this section we recall the category theoretic concept of PILL_Y model. The theory draws heavily on prior work on models of intuitionistic / linear lambda calculi in particular [Barber, 1997, Benton, 1995, Maietti et al., 2005, Maneggia, 2004]. For further details we refer to *loc. cit.*, but also to the note [Møgelberg et al., 2005] which summarises the theory needed for PILL_Y models from the original references.

Since PILL_Y is a language with polymorphism the notion of model is formulated using fibred category theory, and since the category theoretic formulation of models of DILL uses the 2-category of symmetric monoidal categories, we shall use obvious generalisations of the concepts of monoidal category theory to fibred categories.

In this paper we make the general assumption that all fibrations and all fibred structure is split. The assumption of cleavage is similar to the assumption of *chosen* structure in cartesian closed categories, which is necessary to define an interpretation of simply typed lambda calculus. We assume a split cleavage to avoid having to carry isomorphisms around. All examples of LAPL structures that we are aware of are split.

3.1. Definition. *A fibred linear adjunction is a fibred symmetric monoidal adjunction*

$$\begin{tikzcd} \mathbb{C} \arrow[rr, bend right, "G"'] \arrow[dr, "p"'] \arrow[rr, phantom, "\perp"] & & \mathbb{D} \arrow[ll, bend right, "F"'] \arrow[dl] \\ & \mathbb{B}, & \end{tikzcd}$$

where $\mathbb{C}$ is fibred symmetric monoidal closed and the fibred monoidal structure on $\mathbb{D}$ is a fibred cartesian structure.

3.2. Definition. *A fibred linear category is a fibred symmetric monoidal closed category $\mathbb{C} \to \mathbb{B}$ together with a fibred symmetric monoidal comonad $!$ on $\mathbb{C}$ and fibred symmetric monoidal natural transformations $e\colon !(-) \to I$, $d\colon !(-) \to !(-) \otimes !(-)$ such that*

- *For each object A in $\mathbb{C}$, $(!A, e_A, d_A)$ is a commutative comonoid*
- *For each object A in $\mathbb{C}$, e_A, d_A define coalgebra maps from the free coalgebra $\delta_A\colon !A \to !!A$ to the coalgebras $m_I\colon I \to !I$ and*

$$!A \otimes !A \xrightarrow{\delta_A \otimes \delta_A} !!A \otimes !!A \xrightarrow{m} !(!A \otimes !A)$$

respectively, where m_I and m are the comparison functors corresponding to the fibred symmetric monoidal functor $!$.

- *In each fibre, all comonad maps between free coalgebras preserve the comonoid structure.*

In the following we shall often denote a fibred linear category simply by $\mathbb{C} \to \mathbb{B}$ letting the rest of the data being given implicitely.

Given a fibred linear adjunction as in Definition 3.1 the induced fibred comonad on $\mathbb{C}$ gives a fibred linear category. On the other hand there may be several fibred linear adjunctions inducing the same linear category structure. One extreme is letting $\mathbb{D}$ be the co-Eilenberg-Moore category for the comonad, as this is fibred cartesian for any fibred linear category.

From here on we name the categories according to their role in the interpretation of PILL$_Y$. We explain the names later.

3.3. Definition. *A PILL model is a fibred linear category p:* **LinType** $\to$ **Kind** *such that*

- *The category* **Kind** *is cartesian*
- **LinType** $\to$ **Kind** *has a generic object O and, writing Ω for pO, has products with respect to the projections of the form $K \times \Omega \to K$ in* **Kind**. *These products must satisfy the Beck-Chevalley condition.*

A morphism of PILL models from **LinType** $\to$ **Kind** *to* **LinType**$'$ $\to$ **Kind**$'$ *is a strong fibred morphism of fibred symmetric monoidal closed categories*

$$\begin{tikzcd} \mathbf{LinType} \arrow[r, "K"] \arrow[d] & \mathbf{LinType}' \arrow[d] \\ \mathbf{Kind} \arrow[r, "L"] & \mathbf{Kind}' \end{tikzcd}$$

preserving generic object, the cartesian structure in the base categories and products in the fibration.

In the following we shall often talk about a fibred adjunction

$$\begin{tikzcd} \mathbf{LinType} \arrow[rr, bend right, "G"'] \arrow[rr, phantom, "\perp"] \arrow[dr, "p"'] & & \mathbf{Type} \arrow[ll, bend right, "F"'] \arrow[dl] \\ & \mathbf{Kind}, & \end{tikzcd}$$

being a PILL model. By that we shall mean that **LinType** $\to$ **Kind** together with the induced fibred comonad is a PILL model in the sense of Definition 3.3, and moreover, that **Type** is the closure of the fibred coKleisli category for the fibred comonad under finite coproducts inside the coEilenberg-Moore category, which as mentioned above is fibred cartesian. The reason for this is that the category **Type** will turn out to be useful in further developments as we shall see shortly. Notice further that since **Type** is specified there is no ambiguity in this convention. Using this notion of PILL models as fibred adjunctions, specifying **Type** we are following [Maietti et al., 2005].

We sketch how PILL is interpreted in PILL models. Kind contexts are interpreted in **Kind** as $[\![\alpha_1, \ldots, \alpha_n]\!] = \Omega^n$, and a type in kind context $\alpha_1, \ldots, \alpha_n \vdash \sigma$ is interpreted as an object of $\mathbf{LinType}_{\Omega^n}$. A type variable $\alpha_1, \ldots, \alpha_n \vdash \alpha_i$ is interpreted as $\pi_i^*(O)$, where $\pi_i \colon \Omega^n \to \Omega$ is the i'th projection. Polymorphic types are interpreted using products with respect to projections in **Kind** in the usual way. The type constructors $\otimes, \multimap, I$ are interpreted using the fibred symmetric monoidal closed structure and the fibred comonad on **LinType** to interpret !. A term

$$\vec{\alpha} \mid \vec{x} \colon \vec{\sigma}; \vec{x}' \colon \vec{\sigma}' \vdash t \colon \tau$$

is interpreted as a vertical morphism

$$![\![\sigma_1]\!] \otimes \ldots \otimes ![\![\sigma_n]\!] \otimes [\![\sigma_1']\!] \ \otimes \ldots \otimes [\![\sigma_m']\!] \multimap [\![\tau]\!]$$

in **LinType**. Here we have denoted the morphisms in **LinType** by $\multimap$.

Notice how this corresponds to the domain theoretic intuition given earlier for PILL_Y: in domain theory strict continuous maps

$$!A_1 \otimes \ldots \otimes !A_n \otimes B_1 \ \otimes \ldots \otimes B_m \multimap C$$

correspond to continuous maps

$$A_1 \times \ldots \times A_n \times B_1 \ \times \ldots \times B_m \to C$$

strict in each of the B_j variables.

As mentioned earlier, the viewpoint of PILL_Y models as adjunctions is often practical even though in the interpretation given above the category **Type** has not been used. **Type** is useful because it models the purely intuitionistic terms of PILL_Y. To be precise, a PILL_Y-term $\Xi \mid \vec{x} \colon \vec{\sigma}; - \vdash t \colon \tau$ is modeled in **LinType** as a morphism

$$[\![\Xi \mid \vec{x} \colon \vec{\sigma}; - \vdash t \colon \tau]\!] \colon \otimes_i ![\![\Xi \vdash \sigma_i]\!] \multimap [\![\Xi \vdash \tau]\!],$$

and since one can prove $\otimes_i ![\![\Xi \vdash \sigma_i]\!] \cong F(\prod_i G([\![\Xi \vdash \sigma_i]\!]))$, we have, using the fibred adjunction $F \dashv G$, that such a term corresponds to

$$[\![\Xi \mid \vec{x} \colon \vec{\sigma}; - \vdash t]\!]_{\mathbf{Type}} \colon \textstyle\prod_i G([\![\Xi \vdash \sigma_i]\!]) \to G([\![\Xi \vdash \tau]\!])$$

in **Type**.

The interpretation justifies the suggestive notation used in the definition of PILL models: **Kind** models kinds, **LinType** models types and linear maps, and **Type** models types and intuitionistic maps. We shall continue to use similar suggestive notation throughout the paper, to help the reader keep track of the several categories involved in the definition of the concept of LAPL structure.

3.4. Definition. *A PILL_Y model is a PILL model, which models the fixed point operator*

$$Y : \Pi\alpha.\,(\alpha \to \alpha) \to \alpha$$

4. LAPL structures

This section builds on the notion of PILL_Y model to define the notion of LAPL structure, which we will prove is a sound and complete category theoretic notion of model for LAPL. The definition proceeds in two steps. First we define the notion of pre-LAPL structure with notion of admissible relations, which models the fraction of LAPL excluding the relational interpretation of types, i.e., the rule (1). In the second step we identify the extra structure required to interpret the relational interpretation of types, and use this to define the notion of LAPL structure. The second step is described in Section 4.5.

The definition of pre-LAPL structure uses two non-standard notions from fibred category theory, that we explain briefly here. The first concerns a notion of contravariant morphisms of fibrations. We would like to define these to be families of contravariant functors between fibre categories respecting reindexing, just like ordinary maps of fibrations can be described as families of functors respecting reindexing. To make sense of this definition we define a construction, which given a fibration $p \colon \mathbb{E} \to \mathbb{B}$ produces a fibration $p^{\mathrm{fop}} \colon \mathbb{E}^{\mathrm{fop}} \to \mathbb{B}$ where the fibres are the opposite of the fibres of p and reindexing is as in p. Under our standing assumption that all fibrations are split this is easy: by the Grothendieck construction a split fibration p corresponds to a functor $\hat{p} \colon \mathbb{B}^{\mathrm{op}} \to \mathbf{Cat}$, and we define p^{fop} to be the fibration corresponding to the functor $(-)^{\mathrm{op}} \circ \hat{p}$, where $(-)^{\mathrm{op}} \colon \mathbf{Cat} \to \mathbf{Cat}$ is the functor mapping a category to its opposite. In fact p^{fop} can be defined for more general fibrations as described in Appendix A.

The second non-standard notion used in the definition of pre-LAPL structure is the notion of indexed first order logic fibration defined in detail in [Birkedal and Møgelberg, 2005, Appendix A]. This relatively involved categorical notion is needed to model the double indexing occuring here: propositions live in contexts of types and relations which in turn are indexed by contexts of free type variables. We sketch the definitions needed here, but refer to *loc. cit.* for details. An indexed first order logic fibration is a composition of fibrations

$$\mathbb{P} \xrightarrow{r} \mathbb{E} \xrightarrow{q} \mathbb{B}$$

such that each restriction of r to a fibre of q is a first order logic fibration, i.e., a fibred bicartesian closed preorder, which models universal and existential quantification and equality, subject to the usual Beck-Chevalley conditions, but moreover subject to extra Beck-Chevalley conditions corresponding to reindexing in $\mathbb{B}$. We also talk about (r, q) having products and coproducts with respect to classes of maps in $\mathbb{B}$. This means that for every map u in the class in question r has respectively right and left adjoints to reindexing with respect to the cartesian lift of u. These left and right adjoints are subject to two variants of the Beck-Chevalley condition corresponding to maps in $\mathbb{B}$ and $\mathbb{E}$ respectively.

A pre-LAPL structure contains a model of PILL_Y that the logic will reason about. Moreover, it contains a logic fibration modelling the higher order logic present in LAPL. But LAPL is a higher order logic fibration because of the use of relations, and since relations do not exist naturally in PILL_Y models, a pre-LAPL structure contains an extra category called **Ctx**, which the reader should think of as containing the PILL_Y types as well as objects modelling collections of relations. In Definition 4.1 below the collections of objects of relations is given by the functor U, which should be thought of as a contravariant power set functor. The contexts of the mentioned fragment of LAPL are interpreted in **Ctx**, and the propositions of LAPL are modeled in the fibration $\mathbf{Prop} \to \mathbf{Ctx}$.

4.1. Definition. *A **pre-LAPL**-structure is*

1. *a diagram of categories and functors*

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

such that

- *the diagram*

$$\begin{tikzcd} \mathbf{LinType} \arrow[rr, bend right, "G"'] \arrow[rr, phantom, "\perp"] \arrow[dr, "p"'] & & \mathbf{Type} \arrow[ll, bend right, "F"'] \arrow[dl] \\ & \mathbf{Kind} & \end{tikzcd}$$

is a model of PILL_Y.

- *q is a fibration with fibred finite products*
- *(r, q) is an indexed first-order logic fibration [Birkedal and Møgelberg, 2005] which has products and coproducts with respect to projections $K \times \Omega \to K$ in* **Kind** *[Birkedal and Møgelberg, 2005], where Ω again is the p applied to the generic object.*
- *I is a faithful product-preserving map of fibrations.*

2. *a morphism of fibrations:*

$$\begin{tikzcd} (\mathbf{LinType} \times_{\mathbf{Kind}} \mathbf{LinType})^{\mathrm{fop}} \arrow[rr, "U"] \arrow[dr] & & \mathbf{Ctx} \arrow[dl] \\ & \mathbf{Kind} & \end{tikzcd}$$

3. a family of bijections

$$\Psi_{X,A,B} : \mathrm{Hom}_{\mathbf{Ctx}_K}(X, U(A,B)) \to \mathrm{Obj}(\mathbf{Prop}_{X \times I(G(A) \times G(B))})$$

for A and B in $\mathbf{LinType}_K$ and X in $\mathbf{Ctx}_K$ for some K, which

- *is natural in X, A, B with respect to vertical maps*
- *commutes with reindexing functors; that is, if $u : K' \to K$ is a morphism in* **Kind** *and $f : X \to U(A,B)$ is a morphism in $\mathbf{Ctx}_K$, then*

$$\Psi_{u^*(X),u^*(A),u^*(B)}(u^*(f)) = (\bar{u})^*(\Psi_{X,A,B}(f))$$

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

Notice that Ψ is only defined on vertical morphisms.

In this definition $\mathbf{LinType} \times_{\mathbf{Kind}} \mathbf{LinType}$ is defined as the pullback

$$\begin{tikzcd}
\mathbf{LinType} \times_{\mathbf{Kind}} \mathbf{LinType} \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{LinType} \arrow[d, "p"] \\
\mathbf{LinType} \arrow[r, "p"] & \mathbf{Kind}.
\end{tikzcd}$$

Since U is uniquely defined by the requirements on the rest of the structure we will often refer to a pre-LAPL structure simply as the diagram in item 1. Very often we shall omit the indexing of Ψ to ease notation.

We shall use $K, K' \dots$ to denote objects of **Kind**, $A, B, \dots$ to denote objects of **LinType** and $X, Y, \dots$ to denote objects of **Ctx**.

We now explain how to interpret the fragment of LAPL excluding the relational interpretation of types (1) and excluding admissible relations in a pre-LAPL structure. A context

$$\Xi \mid x_1 \colon \sigma_1, \dots x_n \colon \sigma_n \mid R_1 \colon \mathsf{Rel}(\tau_1, \tau_1'), \dots, R_m \colon \mathsf{Rel}(\tau_m, \tau_m')$$

is interpreted in $\mathbf{Ctx}_{[\![\Xi]\!]}$ as

$$\textstyle\prod_{i \le n} IG([\![\sigma_i]\!]) \times \prod_{j \le m} U([\![\tau_j]\!], [\![\tau_j']\!]),$$

where the interpretations of the types is the usual interpretation of types in $\mathbf{LinType} \to \mathbf{Kind}$.

4.2. Lemma. *If $\Xi \mid \Gamma \mid \Theta$ is well formed and $\pi \colon \Omega^{n+1} \to \Omega^n$ is the projection onto the n first coordinates then $[\![\Xi, \alpha \mid \Gamma \mid \Theta]\!] = \pi^*[\![\Xi \mid \Gamma \mid \Theta]\!]$.*

For notational convenience we shall write $[\![\Xi \mid \Gamma \mid \Theta \vdash t \colon \tau]\!]$ for the interpretation of t in **Ctx**, that is for

$$I([\![\Xi \mid \Gamma; - \vdash t \colon \tau]\!]_{\mathbf{Type}}) \circ \pi$$

(note the subscript **Type**), where π is the projection

$$\pi\colon [\![\Xi \mid \Gamma \mid \Theta]\!] \to [\![\Xi \mid \Gamma \mid -]\!]$$

in $\mathbf{Ctx}_{[\![\Xi]\!]}$.

The propositions in the logic are interpreted in **Prop**. Most of the constructions of the logic are interpreted using standard methods from categorical logic, but we present the details here for completeness.

To interpret equality we define

$$[\![\Xi \mid x\colon \tau, y\colon \tau \vdash x =_\tau y]\!] = \coprod\nolimits_{\langle id_{[\![\tau]\!]}, id_{[\![\tau]\!]}\rangle}(\top),$$

where $\coprod_{\langle id_{[\![\tau]\!]}, id_{[\![\tau]\!]}\rangle}$ denotes the left adjoint to reindexing along $\langle id_{[\![\tau]\!]}, id_{[\![\tau]\!]}\rangle\colon [\![\tau]\!] \to [\![\tau]\!] \times [\![\tau]\!]$. Now we can define

$$[\![\Xi \mid \Gamma \mid \Theta \vdash t =_\sigma u]\!] = \langle [\![\Xi \mid \Gamma \mid \Theta \vdash t]\!], [\![\Xi \mid \Gamma \mid \Theta \vdash u]\!]\rangle^* [\![\Xi \mid x\colon \tau, y\colon \tau \vdash x =_\tau y]\!].$$

To interpret $\forall x\colon \sigma_{i_0}.\phi$, recall that a context $\Xi \mid x_1\colon \sigma_1, \ldots, x_n\colon \sigma_n \mid \Theta$ is interpreted as

$$\prod\nolimits_{i\le n} IG[\![\sigma_i]\!] \times [\![\Theta]\!],$$

where $[\![\sigma_i]\!]$ is the usual interpretation of types in **LinType** and the product refers to the fibrewise product in **Ctx**. We may therefore interpret $\forall x\colon \sigma_{i_0}.\phi$ using the right adjoint to reindexing along the projection

$$\pi\colon \prod\nolimits_{i\le n} IG[\![\sigma_i]\!] \times [\![\Theta]\!] \to \prod\nolimits_{i\ne i_0, i\le n} IG[\![\sigma_i]\!] \times [\![\Theta]\!].$$

Likewise, $\forall R\colon \mathsf{Rel}(\sigma,\tau).\,\phi$ is interpreted using right adjoints to reindexing functors related to the appropriate projection in **Ctx**. The existential quantifiers $\exists x\colon \sigma_{i_0}.\phi$ and $\exists R\colon \mathsf{Rel}(\sigma,\tau.)\phi$ are interpreted using left adjoints to the same reindexing functors.

Quantification over types $\forall\alpha.\phi$ and $\exists\alpha.\phi$ is interpreted using respectively right and left adjoints to $\bar\pi^*$ where $\bar\pi$ is the lift of the projection $\pi : [\![\Xi, \alpha\colon \mathsf{Type}]\!] \to [\![\Xi]\!]$ in **Kind** to **Ctx**. The existence of these adjoints is the requirement that (r,q) has products and coproducts. To be more precise, by Lemma 4.2 the cartesian lift of π is a map:

$$\bar\pi\colon [\![\Xi, \alpha \mid \Gamma \mid \Theta]\!] \ \to [\![\Xi \mid \Gamma \mid \Theta]\!]$$

and we define

$$[\![\Xi \mid \Gamma \mid \Theta \vdash \forall\alpha.\,\phi]\!] = \prod\nolimits_{\bar\pi} [\![\Xi,\ \alpha \mid \Gamma \mid \Theta \vdash \phi]\!],$$

where $\prod_{\bar\pi}$ is the right adjoint to $\bar\pi^*$.

Relations with domain σ and codomain τ in contexts $\Xi \mid \Gamma \mid \Theta$ are interpreted as maps from $[\![\Xi \mid \Gamma \mid \Theta]\!]$ into $U([\![\sigma]\!], [\![\tau]\!])$. The relation

$$\Xi \mid \Gamma \mid \Theta, R\colon \mathsf{Rel}(\sigma,\tau) \vdash R\colon \mathsf{Rel}(\sigma,\tau)$$

is interpreted as the projection, and

$$[\![\Xi \mid \Gamma \mid \Theta \vdash (x\colon \sigma, y\colon \tau).\, \phi\colon \mathsf{Rel}(\sigma,\tau)]\!] = \Psi^{-1}([\![\Xi \mid \Gamma, x\colon \sigma, y\colon \tau \mid \Theta \vdash \phi]\!]).$$

We now define the interpretation of $\rho(t,s)$, for a relation ρ and terms t,s of the appropriate types. First, for $\Xi \mid \Gamma \mid \Theta \vdash \rho\colon \mathsf{Rel}(\sigma,\tau)$, we define

$$[\![\Xi \mid \Gamma, x\colon \sigma, y\colon \tau \mid \Theta \vdash \rho(x,y)]\!] = \Psi([\![\Xi \mid \Gamma \mid \Theta \vdash \rho\colon \mathsf{Rel}(\sigma,\tau)]\!]).$$

Next, if $\Xi \mid \Gamma \vdash t\colon \sigma, s\colon \tau$, then

$$\begin{gathered}[\![\Xi \mid \Gamma \mid \Theta \vdash \rho(t,s)]\!] = \\ \langle\langle \pi, \langle [\![\Xi \mid \Gamma \mid \Theta \vdash t]\!], [\![\Xi \mid \Gamma \mid \Theta \vdash s]\!]\rangle\rangle, \pi'\rangle^* [\![\Xi \mid \Gamma, x\colon \sigma, y\colon \tau \mid \Theta \vdash \rho(x,y)]\!],\end{gathered}$$

where π, π' are the projections

$$\pi\colon [\![\Xi \mid \Gamma \mid \Theta]\!] \to [\![\Xi \mid \Gamma]\!] \qquad \pi'\colon [\![\Xi \mid \Gamma \mid \Theta]\!] \to [\![\Xi \mid - \mid \Theta]\!].$$

To interpret admissible relations, we need a little more structure than just a pre-LAPL structure. We need a subfunctor V of U, i.e., a map of fibrations V with domain and codomain as U and a fibred natural transformation $V \Rightarrow U$ whose components are all monomorphic. Thus, for all K in **Kind** and all A, B in $\mathbf{Ctx}_K$, we can consider $V(A,B)$ as a subobject of $U(A,B)$. We think of $V(A,B)$ as the subobject of all admissible relations from A to B (since the isomorphism Ψ allows us to think of $U(A,B)$ as the object of all relations).

Given such a V we can interpret $S\colon \mathsf{AdmRel}(\sigma,\tau)$ as $V([\![\sigma]\!],[\![\tau]\!])$. Admissible relations $\rho\colon \mathsf{AdmRel}(\sigma,\tau)$ are interpreted as maps into $V([\![\sigma]\!],[\![\tau]\!])$. For this to make sense we need, of course, to make sure that the admissible relations in the model in fact contain the relations that are admissible in the logic. We need to assume that of the functor V.

4.3. Definition. *A pre-LAPL structure together with a subfunctor V of U is said to **model admissible relations**, if V is closed under the rules for admissible relations in [Birkedal et al., 2006].*

We say that an implication

$$\Xi \mid \Gamma \mid \Theta \mid \phi_1, \ldots, \phi_n \vdash \psi,$$

in the fragment of LAPL excluding the relational interpretation of types, **holds in** a given pre-LAPL structure with notion of admissible relations, if

$$\bigwedge_{i\le n} [\![\Xi \mid \Gamma \mid \Theta \vdash \phi_i]\!] \le [\![\Xi \mid \Gamma \mid \Theta \vdash \psi]\!]$$

where $\le$ refers to the ordering in the preorder $\mathbf{Prop}_{[\![\Xi|\Gamma|\Theta]\!]}$, and the $\bigwedge$ refers to the intersection in the same preorder.

4.4. Proposition. *For any pre-LAPL structure with given notion of admissible relations the interpretation presented above of the fragment of LAPL excluding the relational interpretation of types is sound in the sense that any implication which is provable in the fragment of the logic also holds in the pre-LAPL structure.*

Most of the proof of Proposition 4.4 is straightforward and very similar to the soundness proof of [Birkedal and Møgelberg, 2005], so we omit it here. Details can be found in [Møgelberg, 2005a].

As a consequence of Proposition 4.4 it makes sense to talk about the internal language of a pre-LAPL structure. In this language we may use types and terms from the PILL_Y model and propositions and relations from the model to form propositions using the constructions of the mentioned fragment of LAPL. We say that an implication of propositions formed in the internal language holds in the internal logic if it holds in the model.

4.5. Semantics of the relational interpretation of types. In this section we define the extra structure needed to interpret the relational interpretation of types. As in Ma & Reynolds work [Ma and Reynolds, 1992] this data is given by a reflexive graph

$$\left(\begin{tikzcd} \mathbf{LinType} \arrow[d] \\ \mathbf{Kind} \end{tikzcd}\right) \begin{array}{c} \longleftarrow \\ \longrightarrow \\ \longleftarrow \end{array} \left(\begin{tikzcd} \mathbf{LinAdmRelations} \arrow[d] \\ \mathbf{AdmRelCtx} \end{tikzcd}\right), \tag{2}$$

this time of PILL models. But unlike in Ma & Reynold's work, we construct the PILL model on the right hand side from the data of the pre-LAPL structure. For more on relations between our approach to models of parametricity and that of Ma & Reynolds, we refer to [Birkedal and Møgelberg, 2005].

We remark that the reason we need this extra structure is that the relational interpretation of types is not just an inductive construction definable only for types of pure PILL_Y. The functor going left to right in (2) will map each type in the PILL_Y model to its relational interpretation.

We now aim to define the fibration $\mathbf{LinAdmRelations} \to \mathbf{AdmRelCtx}$ from a given pre-LAPL structure with a notion of admissible relations. The idea is, that it should be a PILL model in which the objects of the total category are admissible relations in the sense of the pre-LAPL structure. The category $\mathbf{AdmRelCtx}$ is defined as the pullback

$$\begin{tikzcd} \mathbf{AdmRelCtx} \arrow[r] \arrow[d, "\langle \partial_0, \partial_1 \rangle"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{Ctx} \arrow[d] \\ \mathbf{Kind} \times \mathbf{Kind} \arrow[r, "\times"] & \mathbf{Kind}. \end{tikzcd}$$

To define the category $\mathbf{LinAdmRelations}$ we consider the construction of vertical comma category: suppose F is a fibred functor as in

$$\begin{tikzcd} \mathbb{D} \arrow[rr, "F"] \arrow[dr, "p"'] & & \mathbb{E} \arrow[dl, "q"] \\ & \mathbb{B}. & \end{tikzcd} \tag{3}$$

The vertical comma category $V(\mathbb{E} \downarrow F)$ has as objects vertical morphisms triples (E, D, ξ) where E, D are objects in $\mathbb{E}, \mathbb{D}$ respectively and $\xi \colon E \to F(D)$ is a vertical map in $\mathbb{E}$. A morphism from (E, D, ξ) to (E', D', ξ') is a pair of maps $g \colon E \to E', f \colon D \to D'$ in $\mathbb{E}, \mathbb{D}$ respectively such that $\xi' \circ g = F(f) \circ \xi$. Note that f, g need not be vertical. The vertical comma category construction has been studied before, and in fact there is a fibration $V(\mathbb{E} \downarrow F) \to \mathbb{B}$ making it a comma object in the 2-category of fibrations over $\mathbb{B}$ [Jacobs, 1993, Sec 9.4], but that is not important for our application. Instead we need the following.

4.6. Lemma. *Suppose F is a fibred functor as in (3), and denote by $a \colon V(\mathbb{E} \downarrow F) \to \mathbb{D}$ and $b \colon V(\mathbb{E} \downarrow F) \to \mathbb{E}$ the projections. Then b is a fibration and (a, q) is a morphism of fibrations:*

$$\begin{tikzcd} V(\mathbb{E} \downarrow F) \arrow[r, "a"] \arrow[d, "b"'] & \mathbb{D} \arrow[d, "p"] \\ \mathbb{E} \arrow[r, "q"] & \mathbb{B}. \end{tikzcd}$$

Now, the functor V in the definition of pre-LAPL structure induces a fibred functor

$$\begin{tikzcd} (\mathbf{LinType} \times \mathbf{LinType})^{\mathrm{fop}} \arrow[rr, "V'"] \arrow[dr, "(p\times p)^{\mathrm{fop}}"'] & & \mathbf{AdmRelCtx} \arrow[dl] \\ & \mathbf{Kind} \times \mathbf{Kind} & \end{tikzcd}$$

contravariant in each fibre. Using Lemma 4.6, define $\mathbf{LinAdmRelations} \to \mathbf{AdmRelCtx}$ to be $V(\mathbf{AdmRelCtx} \downarrow V')^{\mathrm{fop}} \to \mathbf{AdmRelCtx}$. From Lemma 4.6 we also obtain a fibred functor from $\mathbf{LinAdmRelations} \to \mathbf{AdmRelCtx}$ to $\mathbf{LinType} \times \mathbf{LinType} \to \mathbf{Kind} \times \mathbf{Kind}$. We denote by ∂_0, ∂_1 the two compositions of this fibred functor with the first and second projection respectively:

$$\begin{pmatrix} \mathbf{LinAdmRelations} \\ \downarrow \\ \mathbf{AdmRelCtx} \end{pmatrix} \overset{\partial_0}{\underset{\partial_1}{\rightrightarrows}} \begin{pmatrix} \mathbf{LinType} \\ \downarrow \\ \mathbf{Kind} \end{pmatrix}.$$

Writing out the construction we see that the objects of $\mathbf{AdmRelCtx}$ are triples (K, K', X) where K, K' are objects in $\mathbf{Kind}$ and X is an object of the fibre category $\mathbf{Ctx}_{K\times K'}$. Since such an object can be considered a context in the internal language of the pre-LAPL structure, we write them as $K, K' \mid X$. The fibre of $\mathbf{LinAdmRelations}$ over an object $K, K' \mid X$ has as

Objects: triples (ρ, A, B) where A and B are objects in **LinType** over K and K' respectively and ρ is a vertical map

$$\rho\colon X \to V(\pi^* A, \pi'^* B)$$

in **Ctx**. Here π, π' are first and second projection respectively out of $K \times K'$.

Morphisms: A morphism $(\rho, A, B) \to (\rho', A', B')$ is a pair of morphism

$$(t\colon A \multimap A', u\colon B \multimap B')$$

in $\mathbf{LinType}_K$ and $\mathbf{LinType}_{K'}$ respectively, such that

$$\Psi(\rho) \leq \Psi(V(t,u) \circ \rho')),$$

where we have left the inclusion of V into U implicit.

Expressed in the internal language of the pre-LAPL structure objects of the category **LinAdmRelations** are admissible relations in context, which we shall write as $K, K' \mid X \vdash \rho\colon \mathsf{Rel}(A, B)$ where A is in $\mathbf{LinType}_K$ and B is in $\mathbf{LinType}_{K'}$, and vertical morphisms are pairs of morphisms $t\colon A \multimap A', u\colon B \multimap B'$ preserving relations in the sense that

$$\forall x\colon A, y\colon B\ .\ \rho(x,y) \supset \rho'(t\,x, u\,y)$$

holds. The fibred functors ∂_0, ∂_1 map $K, K' \mid X \vdash \rho\colon \mathsf{Rel}(A, B)$ to A and B respectively.

The following series of lemmas show that the fibration

$$\mathbf{LinAdmRelations} \to \mathbf{AdmRelCtx}$$

is a PILL-model and that the maps ∂_0, ∂_1 are morphisms of PILL-models. The proofs of these lemmas express most of the necessary constructions using LAPL as an internal language. In fact the proofs will show that the PILL-model structure of

$$\mathbf{LinAdmRelations} \to \mathbf{AdmRelCtx}$$

is given by the relational interpretations of the type constructors of PILL_Y, as defined in [Birkedal et al., 2006].

4.7. Lemma. *The fibration* **LinAdmRelations** $\to$ **AdmRelCtx** *has a fibrewise SMCC-structure and the two maps ∂_0, ∂_1 are fibred strict symmetric monoidal functors.*

Proof. The symmetric monoidal closed structure is given by the operations $\otimes, \multimap$ defined on admissible relations in LAPL for $\rho\colon \mathsf{AdmRel}(\sigma, \tau)$, $\rho'\colon \mathsf{AdmRel}(\sigma', \tau')$ as

$$\rho \multimap \rho' = (f\colon \sigma \multimap \sigma', g\colon \tau \multimap \tau').\, \forall x\colon \sigma.\, \forall y\colon \tau.\, \rho(x,y) \supset \rho'(fx, gy).$$

and

$$\rho \otimes \rho' = (f_{\sigma,\tau}, f_{\sigma',\tau'})^*(\forall (\alpha, \beta, R\colon \mathsf{AdmRel}(\alpha, \beta)).\, (\rho \multimap \rho' \multimap R) \multimap R),$$

where $f_{\sigma,\tau} : \sigma \otimes \tau \multimap \prod \alpha.\, (\sigma \multimap \tau \multimap \alpha) \multimap \alpha$ is the canonical map defined as

$$f_{\sigma,\tau}\, x = \text{let } x' \otimes x'' \colon \sigma \otimes \tau \text{ be } x \text{ in } \Lambda\alpha.\, \lambda^\circ h \colon \sigma \multimap \tau \multimap \alpha.\, h\, x'\, x''.$$

The unit for the tensor product is the admissible relation $I_{Rel} \colon \mathsf{AdmRel}(I, I)$ defined as

$$I_{Rel} = (f, f)^*(\forall(\alpha, \beta, R \colon \mathsf{AdmRel}(\alpha, \beta)).\, R \multimap R),$$

where $f \colon I \multimap \prod \alpha.\, \alpha \multimap \alpha$ is defined as $\lambda^\circ x \colon I.\, \text{let } \star \text{ be } x \text{ in } id$, for $id = \Lambda\alpha.\, \lambda^\circ x \colon \alpha.\, x$.

As proved in [Birkedal et al., 2006] these map admissible relations to admissible relations and so do indeed define constructions on **LinAdmRelations** $\to$ **AdmRelCtx**. It is an easy exercise to show that these define functors, and that $\otimes$ defines a symmetric monoidal structure (this simply involves showing that the maps of the symmetric monoidal structure on **LinType** preserve relations).

We show that $\rho \multimap -$ is right adjoint to $- \otimes \rho$. Since we are given a similar adjunction in **LinType**, all we need to show is that

$$(t, s) \colon \rho \multimap (\rho' \multimap \rho'')$$

iff

$$(\hat{t}, \hat{s}) \colon \rho \otimes \rho' \multimap \rho'',$$

where $\hat{t}, \hat{s}$ are the maps corresponding to t, s in the adjunction on **LinType**. Suppose first that

$$(t, s) \colon \rho \multimap (\rho' \multimap \rho'') \text{ and } x(\rho \otimes \rho')y.$$

The definition of the latter says exactly that, for all $(t, s) \colon \rho \multimap (\rho' \multimap \rho'')$, we must have $\rho''(\hat{t}\, x, \hat{s}\, y)$.

Now, suppose $(\hat{t}, \hat{s}) \colon \rho \otimes \rho' \multimap \rho''$ and $x \rho y \wedge x' \rho' y'$. Since $\rho \otimes \rho'(x \otimes x', y \otimes y')$ also $\rho''(\hat{t}(x \otimes x'), \hat{s}(y \otimes y'))$. Hence, since $\hat{t}(x \otimes x') = t\, x\, x'$ (likewise for s), we are done. ∎

4.8. Lemma. *The fibration* **LinAdmRelations** $\to$ **AdmRelCtx** *has a fibred comonad structure. This structure extends to a fibred linear structure, and the maps ∂_0, ∂_1 preserve all this structure on the nose.*

Proof. The operation ! on relations is defined in [Birkedal et al., 2006] as

$$(f_\sigma, f_\tau)^* \forall(\alpha, \beta, R \colon \mathsf{Rel}(\alpha, \beta)).\, (\rho \to R) \multimap R.$$

where $f_\sigma \colon !\sigma \multimap \prod \alpha.\, (\sigma \to \alpha) \multimap \alpha$ is defined as

$$\lambda^\circ x \colon !\sigma.\, \Lambda\alpha.\, \lambda^\circ g \colon \sigma \to \alpha.\, g(x).$$

and for any ρ, ρ' the relation $\rho \to \rho'$ is defined as

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

We need to check that ! defines a functor, i.e., that if $(f,g)\colon \rho \multimap \rho'$, then $(!f, !g)\colon !\rho \multimap !\rho'$. From [Birkedal et al., 2006] we know that it suffices to show $(!f, !g)\colon \rho \to !\rho'$, i.e.,

$$\forall x, y.\, !\rho(!x, !y) \supset !\rho'((!f)(!x), (!g)(!y)).$$

But this holds since $(!f)(!x) = !(f(x))$.

The rest of the proof is a simple check that the counit and comultiplication as well as the underlying structure maps of the fibred linear structure on **LinType** all preserve relations. ∎

4.9. Lemma. *The fibration* **LinAdmRelations** $\to$ **AdmRelCtx** *has products in the base, a generic object and, writing Ω' for the fibration applied to the generic object, simple products with respect to projections in* **AdmRelCtx** *of the form $(K, K' \mid X) \times \Omega' \to (K, K' \mid X)$. The maps ∂_0, ∂_1 preserve this structure.*

Proof. The category **AdmRelCtx** has products:

$$(K_1, K_1' \mid X_1) \times (K_2, K_2' \mid X_2) = K_1 \times K_2, K_1' \times K_2' \mid \pi^*(X_1) \times \pi'^*(X_2)$$

by [Jacobs, 1999, Proposition 9.2.1].

The generic object of the fibration is the triple $(id_{V(\pi^*O,(\pi')^*O)}, O, O)$ where $\pi, \pi'\colon \Omega \times \Omega \to \Omega$ are the projections, since a morphism into that object from $K, K' \mid X$ in **AdmRelCtx** is a pair of maps $(f : K \to \Omega, g : K' \to \Omega)$ in **Kind** together with a morphism from X to $V(\pi^*(f^*O), (\pi')^*(g^*O))$ in $\mathsf{Ctx}_{K\times K'}$.

We now show that the fibration has the mentioned products. The generic object can be expressed in the internal language as $\alpha, \beta \mid - \mid R\colon \mathsf{AdmRel}(\alpha, \beta)$, and so we write a product

$$(K, K' \mid X) \times (\alpha, \beta \mid - \mid R\colon \mathsf{AdmRel}(\alpha, \beta))$$

in **AdmRelCtx** as $(K, \alpha; K', \beta \mid X, R\colon \mathsf{AdmRel}(\alpha, \beta))$. We must show that for any projection in **AdmRelCtx** of the form $\pi\colon (K, \alpha; K', \beta \mid X, R\colon \mathsf{AdmRel}(\alpha, \beta)) \to (K; K' \mid X)$ the weakening functor π^* mapping a relation $K, K' \mid X \vdash \omega\colon \mathsf{AdmRel}(A, A')$ to the relation

$$K, \alpha; K', \beta \mid X, R\colon \mathsf{AdmRel}(\alpha, \beta) \vdash \omega\colon \mathsf{AdmRel}(A, A')$$

has a right adjoint. We define this right adjoint to map a relation

$$K, \alpha; K', \beta \mid X, R\colon \mathsf{AdmRel}(\alpha, \beta) \vdash \rho\colon \mathsf{AdmRel}(B, B')$$

to the relation

$$K, K' \mid X \vdash \forall(\alpha, \beta, R\colon \mathsf{AdmRel}(\alpha, \beta)).\, \rho\colon \mathsf{AdmRel}((\textstyle\prod \alpha\colon \mathsf{Type}.\, B), (\prod \beta\colon \mathsf{Type}.\, B'))$$

defined in [Birkedal et al., 2006] to be

$$(x, y).\, \forall \alpha, \beta\colon \mathsf{Type}.\, \forall R\colon \mathsf{AdmRel}(\alpha, \beta).\, (x\alpha)\rho(y\beta).$$

In the PILL$_Y$ model **LinType** $\to$ **Kind** polymorphic types are modelled using products with respect to projections $K \times \Omega \to \Omega$. This gives for any object A in $\mathbf{LinType}_K$ a bijective correspondence between morphisms $t\colon A \multimap B$ in $\mathbf{LinType}_{K\times\Omega}$ and morphisms $\hat{t}\colon A \multimap \prod\alpha.\, B$ in $\mathbf{LinType}_K$. In fact $\hat{t}$ can be expressed in PILL$_Y$ as

$$K \mid - \vdash \hat{t} = \lambda^\circ x\colon A.\, \Lambda\alpha.\, (t\, x)\colon A \multimap \textstyle\prod\alpha.\, B.$$

Given a relation $K, K' \mid X \vdash \omega\colon \mathsf{AdmRel}(A, A')$ and a pair of morphisms $t\colon A \multimap B$, $u\colon A' \multimap B'$, one can easily check in LAPL that the pair (t, u) preserves relations iff $(\hat{t}, \hat{u})$ does, i.e.,

$$K, \alpha; K', \beta \mid x : A, y : A' \mid X, R\colon \mathsf{AdmRel}(\alpha, \beta) \mid x\omega y \vdash (t\, x)\rho(u\, y)$$

iff

$$K, K' \mid x : A, y : A' \mid X \mid x\omega y \vdash \forall\alpha, \beta\colon \mathsf{Type}.\, \forall R\colon \mathsf{AdmRel}(\alpha, \beta).\, (\hat{t}\, x\, \alpha)\rho(\hat{u}\, y\, \beta),$$

and so the correspondence between (t, u) and $(\hat{t}, \hat{u})$ gives a bijective correspondence between maps $\pi^*\omega \multimap \rho$ in the fibre of **LinAdmRelations** over

$$K, \alpha; K', \beta \mid X, R\colon \mathsf{AdmRel}(\alpha, \beta)$$

and maps $\omega \multimap \forall(\alpha, \beta, R\colon \mathsf{Rel}(\alpha, \beta)).\, \rho$ in the fibre over $K; K' \mid X$ proving that we have in fact defined a product. ∎

The Lemmas 4.7, 4.8, 4.9 collectively state the following.

4.10. Proposition. *The fibration* **LinAdmRelations** $\to$ **AdmRelCtx** *is a PILL model*

This model need not be a PILL$_Y$-model, since for general pre-LAPL structures Y does not necessarily preserve relations.

4.11. Definition. *An **LAPL structure** is a pre-LAPL structure modeling admissible relations, together with a map of PILL-models*

$$\begin{pmatrix} \mathbf{LinType} \\ \downarrow \\ \mathbf{Kind} \end{pmatrix} \xrightarrow{J} \begin{pmatrix} \mathbf{LinAdmRelations} \\ \downarrow \\ \mathbf{AdmRelCtx} \end{pmatrix}$$

extending ∂_0, ∂_1 to a reflexive graph, i.e., $\partial_0 \circ J = \partial_1 \circ J = id$.

Full LAPL can be interpreted in any LAPL structure. To see this we need to show how to interpret the rule

$$\frac{\alpha_1, \ldots, \alpha_n \vdash \sigma(\vec{\alpha})\colon \mathsf{Type} \qquad \Xi \mid \Gamma \mid \Theta \vdash \rho_1\colon \mathsf{AdmRel}(\tau_1, \tau_1'), \ldots, \rho_n\colon \mathsf{AdmRel}(\tau_n, \tau_n')}{\Xi \mid \Gamma \mid \Theta \vdash \sigma[\vec{\rho}]\colon \mathsf{AdmRel}(\sigma(\vec{\tau}), \sigma(\vec{\tau}'))}$$

We first define

$$[\![\vec{\alpha}, \vec{\beta} \mid - \mid \vec{R}\colon \mathsf{AdmRel}(\vec{\alpha}, \vec{\beta}) \vdash \sigma[\vec{R}]]\!] = J([\![\vec{\alpha} \mid \sigma(\vec{\alpha})]\!])$$

Note that this makes sense since J is required to preserve products in the base and generic objects, and so $J([\![\vec{\alpha} \vdash \sigma(\vec{\alpha})]\!])$ is an admissible relation from $\sigma(\vec{\alpha})$ to $\sigma(\vec{\beta})$ in context $[\![\vec{\alpha}; \vec{\beta} \mid \vec{R}\colon \mathsf{AdmRel}(\vec{\alpha}, \vec{\beta})]\!]$. The general $\sigma[\rho]$ is defined as follows: first reindex $J([\![\vec{\alpha} \vdash \sigma(\vec{\alpha})]\!])$ to the right Kind context using

$$\langle [\![\vec{\tau}]\!], [\![\vec{\tau}']\!] \rangle : [\![\Xi]\!] \to \Omega^{2n},$$

thus obtaining

$$[\![\Xi \mid - \mid \vec{R}\colon \mathsf{AdmRel}(\vec{\tau}, \vec{\tau}') \vdash \sigma[\vec{R}]\colon \mathsf{Rel}(\sigma(\vec{\tau}), \sigma(\vec{\tau}'))]\!] = \langle [\![\vec{\tau}]\!], [\![\vec{\tau}']\!] \rangle^* J([\![\vec{\alpha} \vdash \sigma(\vec{\alpha})]\!])$$

and finally we define

$$\begin{gathered} [\![\Xi \mid \Gamma \mid \Theta \vdash \sigma[\vec{\rho}]\colon \mathsf{AdmRel}(\sigma(\vec{\tau}), \sigma(\vec{\tau}'))]\!] = \\ [\![\Xi \mid - \mid \vec{R}\colon \mathsf{AdmRel}(\vec{\tau}, \vec{\tau}') \vdash \sigma[\vec{R}]]\!] \circ [\![\Xi \mid \Gamma \mid \Theta \vdash \vec{\rho}\colon \mathsf{AdmRel}(\vec{\tau}, \vec{\tau}')]\!]. \end{gathered}$$

where by $[\![\Xi \mid \Gamma \mid \Theta \vdash \vec{\rho}\colon \mathsf{AdmRel}(\vec{\tau}, \vec{\tau}')]\!]$ we mean the pairing

$$\langle [\![\Xi \mid \Gamma \mid \Theta \vdash \rho_1]\!], \ldots, [\![\Xi \mid \Gamma \mid \Theta \vdash \rho_n]\!] \rangle.$$

4.12. Remark. Linear Abadi & Plotkin Logic can be modified to include arities of parametricity other than binary. To accomodate such modifications in the definition of LAPL structure, the domain of the functors U and V in the definition of pre-LAPL structure with notion of admissible relations must be changed accordingly.

We define the notion of an implication of LAPL holding in an LAPL structure similarly to the notion of implication holding in a pre-LAPL structure as we did before Proposition 4.4.

4.13. Theorem. [Soundness] *The interpretation given above of LAPL in LAPL structures is sound in the sense that any implication provable in LAPL also holds in any LAPL structure.*

Proof. This theorem extends Proposition 4.4, and like for the proof of Proposition 4.4 we refer to [Møgelberg, 2005a] for details, but do sketch the soundness proof of a few rules.

The rules capturing the inductive definition of relational interpretation of types all hold since J preserves SMCC-structure, generic objects, simple products and !.

Concerning the axiom stating $(\prod\alpha.\, (\alpha \to \alpha) \to \alpha)(Y, Y)$, notice that J is required to be a functor. This means that it maps $[\![Y]\!]\colon I \multimap [\![\prod\alpha.\, (\alpha \to \alpha) \to \alpha]\!]$ to a morphism from I_{Rel} to the relational interpretation of $\prod\alpha.\, (\alpha \to \alpha) \to \alpha$. By the requirement, that $(J, \partial_0, \partial_1)$ is a reflexive graph, this map must be $([\![Y]\!], [\![Y]\!])$. Since $I_{Rel}(\star, \star)$ and $[\![Y]\!](\star) = Y$ we get $(\prod\alpha.\, (\alpha \to \alpha) \to \alpha)(Y, Y)$. ∎

4.14. Completeness. The completeness theorem states an inverse to the soundness theorem, namely that any LAPL implication holding in any LAPL structure is also provable in LAPL. In fact an even stronger result holds:

4.15. Theorem. [Completeness] *There exists an LAPL structure with the property that any formula of LAPL over pure* PILL_Y *holds in this model iff it is provable in LAPL.*

Note that Theorem 4.15 only concerns LAPL as a logic on pure PILL_Y. One can also formulate LAPL as a logic for the internal language of any PILL_Y model, but we do not have a proof of a similar completeness result for this case. The main problem is that we have no way of constructing, for general models of PILL_Y, a relational interpretations as in Section 4.5.

Proof. We construct the LAPL structure syntactically, giving the categories in question the same names as in the diagrams of the definitions of pre-LAPL and LAPL structures. The PILL_Y model in the LAPL structure is a fibred version of the syntactic DILL constructed in [Barber, 1997].

- The category **Kind** has as objects PILL_Y kind contexts Ξ. A morphism from Ξ to $\alpha_1, \ldots, \alpha_n$ is an n-vector of PILL_Y types $(\sigma_1, \ldots, \sigma_n)$ such that all σ_i are well-formed in context Ξ. Composition is by substitution.

- Objects in the fibre of **LinType** over Ξ are well-formed types in this context. Morphisms in this fibre from σ to τ are equivalence classes of terms t such that $\Xi \mid -; x\colon \sigma \vdash t\colon \tau$, under the equivalence relation on PILL_Y terms given by external equality. Composition is by substitution, and reindexing with respect to morphisms in **Kind** is by substitution.

- Objects in the fibre of **Type** over Ξ are well-formed sequences of types in this context. Morphism in this fibre from $\sigma_1, \ldots, \sigma_n$ to $\tau_1, \ldots, \tau_m$ are equivalence classes of sequences of terms $(t_i)_{i \leq m}$, such that for each i the term

$$\Xi \mid \vec{x}\colon \vec{\sigma}; - \vdash t_i\colon \tau_i$$

  is well-formed, under the equivalence relation relating sequences (t_i) and (t'_i) if, for each i, t_i is externally equal to t'_i. Reindexing with respect to morphisms in **Kind** is by substitution.

- The functor **LinType** $\to$ **Type** maps a morphism $-; x\colon \sigma \vdash t\colon \tau$ to $x\colon \sigma; - \vdash t\colon \tau$. The functor going the other way maps a sequence of objects (σ_i) to $\otimes_i !\sigma_i$. It maps a morphism represented by (t_i) from (σ_i) to (τ_i) to the morphism represented by

$$\Xi \mid -; y\colon \otimes_i !\sigma_i \vdash \text{let } \otimes_i x'_i\colon \otimes_i !\sigma_i \text{ be } y \text{ in let } !\vec{x} \text{ be } \vec{x}' \text{ in } \otimes_i !t_i.$$

- The category **Ctx** has as objects in the fibre over Ξ well-formed contexts of LAPL: $\Xi \mid \Gamma \mid \Theta$. A vertical morphism from $\Xi \mid \Gamma \mid \Theta$ to

$$\Xi \mid \Gamma' \mid R_1\colon \mathsf{Rel}(\sigma_1, \tau_1), \ldots, R_n\colon \mathsf{Rel}(\sigma_n, \tau_n), S_1\colon \mathsf{AdmRel}(\sigma'_1, \tau'_1), \ldots, S_m\colon \mathsf{AdmRel}(\sigma'_m, \tau'_m)$$

is an equivalence class of triples, consisting of a vertical morphism $\Xi \mid \Gamma \to \Xi \mid \Gamma'$ in **Type**, a sequence of definable relations $(\rho_1, \ldots, \rho_n)$, and a sequence of admissible relations $(\omega_1, \ldots, \omega_m)$, such that $\Xi \mid \Gamma \mid \Theta \vdash \rho_i \colon \mathsf{Rel}(\sigma_i, \tau_i)$ and $\Xi \mid \Gamma \mid \Theta \vdash \omega_i \colon \mathsf{AdmRel}(\sigma'_i, \tau'_i)$. The equivalence relation relates two such morphisms represented by the same type morphism and the definable relations $(\rho_1, \ldots, \rho_n)$ and $(\rho'_1, \ldots, \rho'_n)$ and admissible relations $(\omega_1, \ldots, \omega_m)$ and $(\omega'_1, \ldots, \omega'_m)$, respectively, if, for each i, j, the formulas $\rho_i \equiv \rho'_i$ and $\omega_j \equiv \omega'_j$ are provable in the logic, where, as usual, $\rho_i \equiv \rho'_i$ is short for

$$\forall x \colon \sigma_i, y \colon \tau_i.\, \rho_i(x_i, y_i) \supset\subset \rho'_i(x_i, y_i),$$

and likewise for $\omega_j \equiv \omega'_j$. The inclusion functor I is the obvious one. Reindexing is by substitution.

- The fibre of the category **Prop** over a context $\Xi \mid \Gamma \mid \Theta$ has as objects equivalence classes of formulas $\Xi \mid \Gamma \mid \Theta \vdash \phi \colon \mathsf{Prop}$, under the equivalence relation given by provable equivalence in LAPL. Each fibre of **Prop** is a preorder, ordered by implication in the logic: $\phi \leq \psi$ iff $\phi \supset \psi$ holds in LAPL. Reindexing is done by substitution: each map in **Ctx** can be written as a composition of a vertical map followed by a cartesian lift of a map in **Kind**, and reindexing with respect to cartesian lifts of morphisms from **Kind** is done by substitution in type-variables, whereas reindexing with respect to vertical maps in **Ctx** is by substitution in term variables and relation variables.

An easy fibred version of the completeness proof in [Barber, 1997] shows that **Kind**, **Type**, **LinType** together with the functors described above form a PILL$_Y$ model. The fibration **Ctx** $\to$ **Kind** clearly has fibred products formed by appending contexts, and the inclusion functor I is clearly faithful and product-preserving.

We need to prove that **Prop** $\to$ **Ctx** $\to$ **Kind** is an indexed first-order logic fibration with products and coproducts with respect to simple projections in **Kind**. The fibrewise bicartesian structure is given by $\vee, \wedge, \supset, \bot, \top$. Fibred simple products and coproducts are given by quantifying over relations and variables, simple products in the composite fibration is given by quantifying over types. We can in fact prove that the composite fibration has all indexed products and coproducts (in particular, that it has equality).

Suppose $(\vec{t}, \vec{\rho})$ represents a vertical morphism from $\Xi \mid \vec{x} \colon \vec{\sigma} \mid \vec{R}$ to $\Xi \mid \vec{y} \colon \vec{\tau} \mid \vec{S}$ in **Ctx** (the vectors $\vec{R}, \vec{S}$ consist of both relations and admissible relations, and the vector $\vec{\rho}$ is a concatenation of the corresponding vectors of relations and admissible relations from the definition above). We can then define the product functor in **Prop** as:

$$\begin{gathered}\textstyle\prod_{(\vec{t},\vec{\rho})}(\Xi \mid \vec{x} \mid \vec{R} \vdash \phi(\vec{x}, \vec{R})) = \\ \Xi \mid \vec{y} \mid \vec{S} \vdash \forall \vec{x}.\, \forall \vec{R}(\vec{t}\vec{x} = \vec{y} \wedge (\vec{\rho}(\vec{x}, \vec{R}) \equiv \vec{S}) \supset \phi(\vec{x}, \vec{R})).\end{gathered}$$

We define coproduct as:

$$\begin{gathered}\textstyle\coprod_{(\vec{t},\vec{\rho})}(\Xi \mid \vec{x} \mid \vec{R} \vdash \phi(\vec{x}, \vec{R})) = \\ \Xi \mid \vec{y} \mid \vec{S} \vdash \exists \vec{x}.\, \exists \vec{R}.\, \vec{t}\vec{x} = \vec{y} \wedge \vec{\rho}(\vec{x}, \vec{R}) \equiv \vec{S} \wedge \phi(\vec{x}, \vec{R})).\end{gathered}$$

We remark that the special case of equality in the model is the obvious

$$(\Xi \mid \Gamma, x\colon \sigma \mid \Theta \vdash \phi) \mapsto (\Xi \mid \Gamma, x\colon \sigma, y\colon \sigma \mid \Theta \vdash \phi \wedge x =_\sigma y)$$

The fibred functor U of item 2 in Definition 4.1 is defined as

$$U(\sigma,\tau) = R\colon \mathsf{Rel}(\sigma,\tau)$$

and

$$U(t\colon \sigma \multimap \sigma', u\colon \tau \multimap \tau') = \Xi \mid R\colon \mathsf{Rel}(\sigma',\tau') \vdash (x\colon \sigma, y\colon \tau).\, R(tx, uy),$$

where x, y are fresh variables.

The required isomorphism Ψ maps $\Xi \mid \Gamma \mid \Theta \vdash \rho\colon \mathsf{Rel}(\sigma,\tau)$ to $\Xi \mid \Gamma, x\colon \sigma, y\colon \tau \mid \Theta \vdash \rho(x,y)$. The functor V is defined as

$$V(\sigma,\tau) = R\colon \mathsf{AdmRel}(\sigma,\tau)$$

and

$$V(t\colon \sigma \multimap \sigma', u\colon \tau \multimap \tau') = \Xi \mid R\colon \mathsf{AdmRel}(\sigma',\tau') \vdash (x\colon \sigma, y\colon \tau).\, R(tx, uy).$$

We have defined a pre-LAPL structure modeling admissible relations. If we construct **AdmRelCtx** as in the definition of LAPL structure, we obtain:

Objects: $\vec{\alpha}, \vec{\beta} \mid \Gamma \mid \vec{R}\colon \mathsf{AdmRel}(\vec{\sigma}(\vec{\alpha}), \vec{\tau}(\vec{\beta})), \vec{R}'\colon \mathsf{Rel}(\vec{\sigma}'(\vec{\alpha}), \vec{\tau}'(\vec{\beta}))$.

Morphisms: A morphism from

$$\vec{\alpha}, \vec{\beta} \mid \Gamma \mid \vec{R}\colon \mathsf{AdmRel}(\vec{\sigma}(\vec{\alpha}), \vec{\tau}(\vec{\beta})), \vec{R}'\colon \mathsf{Rel}(\vec{\sigma}'(\vec{\alpha}), \vec{\tau}'(\vec{\beta}))$$

to

$$\vec{\alpha}', \vec{\beta}' \mid \Gamma' \mid \vec{S}\colon \mathsf{AdmRel}(\vec{\omega}(\vec{\alpha}'), \vec{\kappa}(\vec{\beta}')), \vec{S}'\colon \mathsf{Rel}(\vec{\omega}'(\vec{\alpha}'), \vec{\kappa}'(\vec{\beta}'))$$

consists of two morphism in **Kind**:

$$\vec{\mu} : \vec{\alpha} \to \vec{\alpha}'$$

and

$$\vec{\nu} : \vec{\beta} \to \vec{\beta}',$$

a morphism from $\vec{\alpha}, \vec{\beta} \mid \Gamma$ to $\vec{\alpha}, \vec{\beta} \mid \Gamma'[\vec{\mu}, \vec{\nu}/\vec{\alpha}', \vec{\beta}']$ in $\mathbf{LinType}_{\vec{\alpha},\vec{\beta}}$, and a sequence of admissible relations $\vec{\rho}$ and a sequence of relations $\vec{\rho}'$ such that, for all i,j,

$$\begin{gathered}\vec{\alpha}, \vec{\beta} \mid \Gamma \mid \vec{R}\colon \mathsf{AdmRel}(\vec{\sigma}(\vec{\alpha}), \vec{\tau}(\vec{\beta})), \vec{R}'\colon \mathsf{Rel}(\vec{\sigma}'(\vec{\alpha}), \vec{\tau}'(\vec{\beta})) \vdash \rho_i\colon \mathsf{AdmRel}(\omega_i(\vec{\mu}), \kappa_i(\vec{\nu}))\\ \vec{\alpha}, \vec{\beta} \mid \Gamma \mid \vec{R}\colon \mathsf{AdmRel}(\vec{\sigma}(\vec{\alpha}), \vec{\tau}(\vec{\beta})), \vec{R}'\colon \mathsf{Rel}(\vec{\sigma}'(\vec{\alpha}), \vec{\tau}'(\vec{\beta})) \vdash \rho_j'\colon \mathsf{Rel}(\omega_j'(\vec{\mu}), \kappa_j'(\vec{\nu})).\end{gathered}$$

As in **Ctx** these morphisms are identified up to provable equivalence of the relations.

The fibre of **LinAdmRelations** over an object

$$\vec{\alpha}, \vec{\beta} \mid \Gamma \mid R\colon \mathsf{AdmRel}(\vec{\sigma}(\vec{\alpha}), \vec{\tau}(\vec{\beta})), \vec{R}'\colon \mathsf{Rel}(\vec{\sigma}'(\vec{\alpha}), \vec{\tau}'(\vec{\beta}))$$

in **AdmRelCtx** becomes:

Objects: Equivalence classes of admissible relations

$$\vec{\alpha}, \vec{\beta} \mid \vec{\Gamma} \mid R\colon \mathsf{AdmRel}(\vec{\sigma}(\vec{\alpha}), \vec{\tau}(\vec{\beta})), \vec{R}'\colon \mathsf{Rel}(\vec{\sigma}'(\vec{\alpha}), \vec{\tau}'(\vec{\beta})) \vdash \rho\colon \mathsf{AdmRel}(\sigma(\vec{\alpha}), \tau(\vec{\beta})).$$

Morphisms: A morphism from $\rho\colon \mathsf{AdmRel}(\sigma(\vec{\alpha}), \tau(\vec{\beta}))$ to $\rho'\colon \mathsf{AdmRel}(\sigma'(\vec{\alpha}), \tau'(\vec{\beta}))$ is a pair of morphisms $t : \sigma \multimap \sigma'$, $u : \tau \multimap \tau'$ such that it is provable in the logic that:

$$\forall x\colon \sigma. \forall y\colon \tau. \rho(x, y) \supset \rho'(tx, uy)$$

We will construct the map J as a map of fibred linear categories from **LinType** $\to$ **Kind** to **LinAdmRelations** $\to$ **AdmRelCtx** as follows. On the base categories J is defined on objects as

$$J(\alpha_1, \ldots, \alpha_n) = \alpha_1, \ldots, \alpha_n; \beta_1, \ldots, \beta_n \mid R_1\colon \mathsf{AdmRel}(\alpha_1, \beta_1), \ldots, R_n\colon \mathsf{AdmRel}(\alpha_n, \beta_n),$$

where the $\vec{\beta}$ are fresh variables. We define J on the objects of the total categories (and on the morphisms of the base category) as

$$J(\vec{\alpha} \vdash \sigma\colon \mathsf{Type}) = \vec{\alpha}, \vec{\beta} \mid \vec{R}\colon \mathsf{AdmRel}(\vec{\alpha}, \vec{\beta}) \vdash \sigma[R]\colon \mathsf{AdmRel}(\sigma(\vec{\alpha}), \sigma(\vec{\beta})).$$

To define J on morphisms of the fibre categories, suppose $\vec{\alpha} \mid -; - \vdash t\colon \sigma \multimap \tau$. We define $J(t) = (t, t)$. To see that (t, t) in fact is a map from $\sigma[\vec{R}]$ to $\tau[\vec{R}]$, notice that the Logical Relations Lemma [Birkedal et al., 2006, Lemma 1.39] tells us that

$$\Lambda \vec{\alpha}.\, t (\textstyle\prod \vec{\alpha}.\, \sigma \multimap \tau) \Lambda \vec{\alpha}.\, t,$$

which means exactly that $(t, t)\colon \sigma[\vec{R}] \multimap \tau[\vec{R}]$.

The rules for the relational interpretation of types tell us that J is a strict fibred symmetric monoidal closed functor preserving products and ! on the nose. Since the counit and comultiplication of the fibred comonad on **LinAdmRelations** $\to$ **AdmRelCtx** are simply (ϵ, ϵ) and (δ, δ), for ϵ and δ the counit and comultiplication of **LinType** $\to$ **Kind** respectively, it is clear that J preserve these as well.

Now, by definition, a formula holds in this LAPL structure iff it is provable LAPL. ∎

5. Parametric LAPL structures

This section defines the notion of parametric LAPL structure. These are LAPL structures in which the identity extension schema holds and which satisfy a condition called very strong equality. Identity extension allows us to reason using parametricity in the internal logic of the LAPL structure, and very strong equality allows us to transfer the results from the internal logic to results about the PILL_Y model. The need for very strong equality is perhaps best illustrated by an extreme case: suppose we are given an LAPL structure in which $\mathbf{Prop} = \mathbf{Ctx}$ and the fibration $r\colon \mathbf{Prop} \to \mathbf{Ctx}$ is the identity functor. In this case the fibres of the fibration are all trivial, and so any proposition in LAPL holds in such a model. In particular, being able to prove that two vertical morphism of **LinType** with the same domain and codomain are equal in the internal language of the LAPL structure says nothing about them being equal morphisms in **LinType**.

5.1. Definition. *A preorder fibration $\mathbb{E} \to \mathbb{B}$ modelling equality has very strong equality if for every two morphisms f, g in $\mathbb{B}$ with same domain and codomain, $f = g$ holds in the logic of the fibration only if f and g are in fact equal. An LAPL structure has very strong equality if each restriction of $r\colon \mathbf{Prop} \to \mathbf{Ctx}$ to a fibre of $q\colon \mathbf{Ctx} \to \mathbf{Kind}$ has very strong equality.*

As argued in [Birkedal and Møgelberg, 2005] very strong equality implies that the two extensionality schemes

$$\forall \alpha, \beta\colon \mathbf{Type}.\, \forall f, g\colon \alpha \multimap \beta.\, (\forall x\colon \alpha.\, f(x) =_\tau g(x)) \supset f =_{\alpha\multimap\beta} g$$
$$\forall x, y\colon \textstyle\prod \alpha.\, \sigma.\, (\forall \alpha\colon \mathsf{Type}.\, x\, \alpha =_\sigma y\, \alpha) \supset x =_{\prod \alpha.\sigma} y$$

hold in the internal logic of the LAPL structure.

5.2. Definition. *A **parametric LAPL structure** is an LAPL structure with very strong equality in which identity extension holds in the internal logic.*

5.3. Solving recursive domain equations in parametric LAPL structures. This section identifies the semantic notion of recursive domain equation that can be solved using parametricity in LAPL structures. This can be also seen as an example of how we can reason in the internal logic of parametric LAPL structures and use very strong equality to transfer these results to results about the PILL_Y model in question.

Below we shall use the notation A^B for the closed structure in symmetric monoidal closed categories. This is meant to simplify otherwise lengthy notation.

5.4. Definition. *An endofunctor $F : \mathbb{B}^{\mathrm{op}} \times \mathbb{B} \to \mathbb{B}$, for $\mathbb{B}$ a linear category, is called **strong** if there exists a natural transformation $t_{A,B,A',B'} : !(A^{A'}) \otimes !((B')^B) \multimap F(A', B')^{F(A,B)}$ preserving identity and composition:*

$$\begin{tikzcd} I \arrow[rr, "!\widehat{id_A} \otimes !\widehat{id_B}"] \arrow[drr, "\widehat{id_{FA}}"'] & & !(A^A) \otimes !(B^B) \arrow[d, "t_{A,B,A,B}"] \\ & & F(A,B)^{F(A,B)} \end{tikzcd} \qquad \begin{tikzcd} !(A^{A'}) \otimes !((B')^B) \otimes !((A')^{A''}) \otimes !((B'')^{B'}) \arrow[r, "comp"] \arrow[d, "t \otimes t"'] & !(A^{A''}) \otimes !((B'')^B) \arrow[d, "t"] \\ F(A', B')^{F(A,B)} \otimes F(A'', B'')^{F(A',B')} \arrow[r, "comp"] & F(A'', B'')^{F(A,B)}. \end{tikzcd}$$

*The natural transformation t is called the **strength** of the functor F.*

One should note that t in the definition above represents the morphism part of the functor F in the sense that it makes the diagram

$$
\begin{tikzcd}
I \arrow[r, "!\widehat{f}\otimes!\widehat{g}"] \arrow[dr, "\widehat{F(f,g)}"'] & !(A^{A'})\otimes!((B')^{B}) \arrow[d, "t_{A,B,A',B'}"] \\
& F(A',B')^{F(A,B)}
\end{tikzcd}
\tag{4}
$$

commute, for any pair of morphisms $f\colon A' \multimap A, g\colon B \multimap B'$. This follows from the commutative diagram

$$
\begin{tikzcd}
I \arrow[dr, "!\widehat{id}\otimes!\widehat{id}"] \arrow[drr, bend left, "\widehat{id}"] \arrow[ddr, bend right, "!\widehat{f}\otimes!\widehat{g}"'] & & \\
& !A^{A}\otimes!B^{B} \arrow[r, "t"] \arrow[d, "!(A^{f})\otimes!(g^{B})"] & F(A,B)^{F(A,B)} \arrow[d, "F(f,g)^{F(A,B)}"] \\
& !A^{A'}\otimes!(B')^{B} \arrow[r, "t"] & F(A',B')^{F(A,B)}.
\end{tikzcd}
$$

In fact, (4) shows that we can define the action of F on morphisms from the strength.

5.5. Definition. *If $\mathbb{E} \to \mathbb{B}$ is a fibred SMCC, then a fibred functor*

$$
\begin{tikzcd}
\mathbb{E}^{\mathrm{fop}} \times_{\mathbb{B}} \mathbb{E} \arrow[rr, "F"] \arrow[dr] & & \mathbb{E} \arrow[dl] \\
& \mathbb{B} &
\end{tikzcd}
$$

*is called **strong fibred** if there exists a fibred natural transformation t from*

$$!((=)^{(-)})\otimes!((=')^{(-')}) \text{ to } F(-,=')^{F(=,-')}$$

*satisfying commutativity of the two diagrams of Definition 5.4 in each fibre. The natural transformation t is called the **strength** of the functor F.*

The fibred strength t is a natural transformation between two fibred functors

$$
\begin{tikzcd}
\mathbb{E} \times_{\mathbb{B}} \mathbb{E}^{\mathrm{fop}} \times_{\mathbb{B}} \mathbb{E}^{\mathrm{fop}} \times_{\mathbb{B}} \mathbb{E} \arrow[rr] \arrow[dr] & & \mathbb{E} \arrow[dl] \\
& \mathbb{B} &
\end{tikzcd}
$$

For example, in the case of a PILL$_Y$-model, the interpretation of any inductively constructed type $\alpha,\beta \vdash \sigma(\alpha,\beta)$ with α occurring only negatively and β only positively induces a strong fibred functor, since as described in [Birkedal et al., 2006], for each such type we can define a closed term

$$t\colon \prod \alpha, \beta, \alpha', \beta'.\, (\alpha' \multimap \alpha) \to (\beta \multimap \beta') \to \sigma(\alpha, \beta) \multimap \sigma(\alpha', \beta')$$

The object part of the functor F is then defined as

$$F(A, B) = [\![\sigma(A, B)]\!]$$

and the strength s of the functor is defined as

$$(s_K)_{A,A',B,B'} = [\![K \mid -; - \vdash t\; A\; A'\; B\; B']\!].$$

The morphism part of the functor is induced by the strength.

In fact, in a sense these are the only strong fibred functors on PILL-models.

5.6. Lemma. *Suppose*

$$\begin{tikzcd} \mathbf{LinType}^{\mathrm{fop}} \times_{\mathbf{Kind}} \mathbf{LinType} \arrow[rr, "F"] \arrow[dr] & & \mathbf{LinType} \arrow[dl] \\ & \mathbf{Kind} & \end{tikzcd}$$

is a strong fibred functor on a PILL-model. Then there exists a type $\alpha, \beta \vdash \sigma$ and a closed term

$$t\colon \prod \alpha, \beta, \alpha', \beta'.\, (\alpha' \multimap \alpha) \to (\beta \multimap \beta') \to \sigma(\alpha, \beta) \multimap \sigma(\alpha', \beta')$$

in the internal language of $\mathbf{LinType} \to \mathbf{Kind}$ *inducing F.*

Proof. Let $O\colon \mathbf{LinType}_\Omega$ denote the generic object of the fibration $\mathbf{LinType} \to \mathbf{Kind}$. For each type $A\colon \mathbf{LinType}_K$ there exists a map $K \to \Omega$, which we will denote $\hat{A}$ such that $\hat{A}^* O = A$.

Define $\sigma = F([\![\alpha, \beta \vdash \alpha]\!], [\![\alpha, \beta \vdash \beta]\!])$. Then

$$F(A, A') = F(\langle \hat{A}, \hat{A}' \rangle^* ([\![\alpha, \beta \vdash \alpha]\!], [\![\alpha, \beta \vdash \beta]\!])) = \langle \hat{A}, \hat{A}' \rangle^* \sigma$$

for any pair of types $(A, A') \in (\mathbf{LinType}^{\mathrm{fop}} \times_{\mathbf{Kind}} \mathbf{LinType})_K$, since F is fibred. In the internal language $K \vdash \sigma(A, A')$ is interpreted as $\langle \hat{A}, \hat{A}' \rangle^* \sigma$ and so indeed σ induces the action of F on objects.

Let s denote the strength of the fibred functor F. Consider the component

$$(s_{\Omega^4})_{[\![\alpha,\beta,\alpha',\beta' \vdash \alpha]\!], [\![\alpha,\beta,\alpha',\beta' \vdash \beta]\!], [\![\alpha,\beta,\alpha',\beta' \vdash \alpha']\!], [\![\alpha,\beta,\alpha',\beta' \vdash \beta']\!]}$$

and denote it by t'. In the internal language, t' is a term, and we can consider the polymorphic term

$$- \vdash t = \Lambda\alpha.\, \Lambda\beta.\, \Lambda\alpha'.\, \Lambda\beta'.\, t' \colon \prod \alpha, \beta, \alpha', \beta'.\, (\alpha' \multimap \alpha) \to (\beta \multimap \beta') \to (\sigma(\alpha, \beta) \multimap \sigma(\alpha', \beta')).$$

We just need to show that the strength induced by the term t is in fact s, but

$$[\![K \vdash t\; A\; A'\; B\; B']\!] = \langle \hat{A}, \hat{A}', \hat{B}, \hat{B}' \rangle^* t' = (s_K)_{A,A',B,B'}$$

since s is preserved by reindexing. ∎

5.7. Theorem. *In a parametric LAPL structure, for any strong fibred functor F there exists a closed type rec $\alpha. F(\alpha, \alpha)$ such that*

$$F(rec\ \alpha. F(\alpha,\alpha), rec\ \alpha. F(\alpha,\alpha)) \cong rec\ \alpha. F(\alpha,\alpha)$$

in $\mathbf{LinType}_1$ *for* 1 *the terminal object of* $\mathbf{Kind}$. *The isomorphism is an initial dialgebra for the functor* $F_1 \colon {\mathbf{LinType}_1}^{\mathrm{op}} \times \mathbf{LinType}_1 \to \mathbf{LinType}_1$.

Moreover, for any object K in $\mathbf{Kind}$,

$$F(!_K^*(rec\ \alpha. F(\alpha,\alpha)), !_K^*(rec\ \alpha. F(\alpha,\alpha))) \cong !_K^*(rec\ \alpha. F(\alpha,\alpha)),$$

where $!_K$ is the unique map $K \to 1$ in $\mathbf{Kind}$, *holds in* $\mathbf{LinType}_K$ *and is an initial dialgebra for F_K.*

Proof. Since by Lemma 5.6 we can express F in the internal language, the construction of solutions to recursive domain equations and the proofs of correctness in [Birkedal et al., 2006] goes through in the internal language of the LAPL structure, if we substitute the functorial interpretation of types as constructed in PILL_Y by the polymorphic term provided by Lemma 5.6. This gives the closed type rec $\alpha. F(\alpha,\alpha)$ and the isomorphism $F(\mathrm{rec}\ \alpha. F(\alpha,\alpha), \mathrm{rec}\ \alpha. F(\alpha,\alpha)) \cong \mathrm{rec}\ \alpha. F(\alpha,\alpha)$ provable in LAPL. Finally, by very strong equality the isomorphism $F(\mathrm{rec}\ \alpha. F(\alpha,\alpha), \mathrm{rec}\ \alpha. F(\alpha,\alpha)) \cong \mathrm{rec}\ \alpha. F(\alpha,\alpha)$ not only holds in LAPL but also in the category $\mathbf{LinType}_1$.

The rest of the theorem can be proved likewise. ∎

5.8. Parametrized recursive type equations. An inductively constructed type $\alpha_1, \beta_1, \ldots, \alpha_n, \beta_n \vdash \sigma$ with $2n$ free type variables, in which the variables $\vec{\alpha}$ occur only negatively and the variables $\vec{\beta}$ only positively induces a fibred functor

$$\begin{tikzcd}
(\mathbf{LinType}^{\mathrm{fop}} \times_{\mathbf{Kind}} \mathbf{LinType})^n \arrow[rr, "F"] \arrow[dr] & & \mathbf{LinType} \arrow[dl] \\
& \mathbf{Kind}. &
\end{tikzcd}$$

On the other hand Definition 5.5 can easily be extended to define what it means that a functor F as above is strong fibred, and Lemma 5.6 extends to show that such strong fibred functors correspond to types σ as above and closed polymorphic terms of type

$$\prod \vec{\alpha}, \vec{\beta}, \vec{\alpha}', \vec{\beta}'. (\alpha_1' \multimap \alpha_1) \to (\beta_1 \multimap \beta_1') \to \ldots \to (\alpha_n' \multimap \alpha_n) \to (\beta_n \multimap \beta_n') \to \sigma(\vec{\alpha}, \vec{\beta}) \multimap \sigma(\vec{\alpha}', \vec{\beta}')$$

in the internal language. The following theorem is then the corresponding extension of Theorem 5.7 obtained using the analysis of [Birkedal et al., 2006].

5.9. Theorem. *In a parametric LAPL structure, for any strong fibred functor*

$$\begin{tikzcd}
(\mathbf{LinType}^{\mathrm{fop}} \times_{\mathbf{Kind}} \mathbf{LinType})^{n+1} \arrow[rr, "F"] \arrow[dr] & & \mathbf{LinType} \arrow[dl] \\
& \mathbf{Kind}. &
\end{tikzcd}$$

there exists a strong fibred functor

$$\begin{tikzcd}
(\mathbf{LinType}^{\mathrm{fop}} \times_{\mathbf{Kind}} \mathbf{LinType})^n \arrow[rr, "FixF"] \arrow[dr] & & \mathbf{LinType} \arrow[dl] \\
& \mathbf{Kind}. &
\end{tikzcd}$$

and a fibred natural isomorphism

$$\begin{array}{rl} & F(A_1, B_1, \ldots, A_n, B_n, FixF(B_1, A_1, \ldots, B_n, A_n), FixF(A_1, B_1, \ldots, A_n, B_n)) \\ \cong & FixF(A_1, B_1, \ldots, A_n, B_n). \end{array}$$

Moreover, if G is a strong fibred functor

$$\begin{tikzcd}
(\mathbf{LinType}^{\mathrm{fop}} \times_{\mathbf{Kind}} \mathbf{LinType})^m \arrow[rr, "G"] \arrow[dr] & & (\mathbf{LinType}^{\mathrm{fop}} \times_{\mathbf{Kind}} \mathbf{LinType})^n \arrow[dl] \\
& \mathbf{Kind}. &
\end{tikzcd}$$

then $Fix(F \circ (G \times id)) = FixF \circ G$.

The universal condition satisfied by the functor $\mathrm{Fix}F$ is a parametrized version of dinaturality. It has been described in [Birkedal et al., 2006], but for completeness we repeat it here in a semantic formulation.

For any object $K \in \mathbf{Kind}$, all objects $A_1, \ldots A_n, B_1, \ldots, B_n, C, C'$ in the fibre $\mathbf{LinType}_K$ and all morphisms

$$\begin{array}{l} g\colon F(A_1, B_1, \ldots, A_n, B_n, C', C) \multimap C \\ g'\colon C' \multimap F(B_1, A_1, \ldots, B_n, A_n, C, C') \end{array}$$

also in the fibre $\mathbf{LinType}_K$, there exist unique maps h, h' making the diagrams

$$\begin{tikzcd}
F(A_1, B_1, \ldots, \mathrm{Fix}F(B_1, A_1, \ldots), \mathrm{Fix}F(A_1, B_1, \ldots)) \arrow[r, "\cong", -o] \arrow[d, "{F(A_1,B_1,\ldots,A_n,B_n,h',h)}"', -o] & \mathrm{Fix}F(A_1, B_1, \ldots, A_n, B_n) \arrow[d, "h", -o] \\
F(A_1, B_1, \ldots, A_n, B_n, C, C) \arrow[r, "g", -o] & C
\end{tikzcd}$$

$$\begin{tikzcd}
C' \arrow[r, "g'", -o] \arrow[d, "h'"', -o] & F(B_1, A_1, \ldots, B_n, A_n, C, C') \arrow[d, "{F(B_1,A_1,\ldots,B_n,A_n,h,h')}", -o] \\
F(B_1, A_1, \ldots, \mathrm{Fix}F(A_1, B_1, \ldots), \mathrm{Fix}F(B_1, A_1, \ldots)) \arrow[r, "\cong", -o] & \mathrm{Fix}F(B_1, A_1, \ldots, B_n, A_n)
\end{tikzcd}$$

commute in the fibre category $\mathbf{LinType}_K$.

6. Conclusions

We have defined the category-theoretic notion of LAPL structure, and proved that this is a sound and complete notion of model for LAPL, a logic for parametricity and domain theory. We propose that parametric LAPL structures be used as a notion of domain theoretic models of parametric polymorphism. This notion is useful, because, as we have showed, semantic versions of the consequences for parametricity provable logically hold in parametric LAPL structures.

In forthcoming papers we show that the notion of parametric LAPL structure is also a general notion of domain theoretic models of parametricity, by giving a series of examples of parametric LAPL structures. These examples include a model constructed using partial equivalence relations over a domain theoretic model of the untyped lambda calculus, Simpson and Rosolini's construction of models in intuitionistic set theory, a model constructed using the syntax of Lily and a general parametric completion process as in [Robinson and Rosolini, 1994]. In all these cases the notion of admissible relation is interpreted differently, and a central part of each of these verifications, is to show that the concrete notions of admissible relations satisfy the axioms of [Birkedal et al., 2006].

Acknowledgments

We gratefully acknowledge discussions with Shin-ya Katsumata, Milly Maietti, Gordon Plotkin, John Reynolds, Pino Rosolini and Alex Simpson. Thanks also to the two anonymous referees for helpful suggestions.

References

Barber, A. (1997). *Linear Type Theories, Semantics and Action Calculi.* PhD thesis, Edinburgh University.

Benton, P. (1995). A mixed linear and non-linear logic: Proofs, terms and models (preliminary report). Technical report, University of Cambridge.

Bierman, G. M., Pitts, A. M., and Russo, C. V. (2000). Operational properties of Lily, a polymorphic linear lambda calculus with recursion. In *Fourth International Workshop on Higher Order Operational Techniques in Semantics, Montréal*, volume 41 of *Electronic Notes in Theoretical Computer Science*. Elsevier.

Birkedal, L. and Møgelberg, R. E. (2005). Categorical models of Abadi-Plotkin's logic for parametricity. *Mathematical Structures in Computer Science*, 15(4):709–772.

Birkedal, L., Møgelberg, R. E., and Petersen, R. L. (2006). Linear Abadi & Plotkin logic. *Logical Methods in Computer Science*, 2.

Birkedal, L., Møgelberg, R. E., and Petersen, R. L. (2007). Domain-theoretical models of parametric polymorphism. *Theor. Comput. Sci*, 388(1-3):152–172.

Freyd, P. (1990a). Algebraically complete categories. In Carboni, A., Pedicchio, M. C., and Rosolini, G., editors, *Category Theory. Proceedings, Como 1990*, volume 1488 of *Lecture Notes in Mathematics*, pages 95–104. Springer-Verlag.

Freyd, P. (1990b). Recursive types reduced to inductive types. In *Proceedings of the fifth IEEE Conference on Logic in Computer Science*, pages 498–507.

Freyd, P. (1991). Remarks on algebraically compact categories. In Fourman, M. P., Johnstone, P., and Pitts, A. M., editors, *Applications of Categories in Computer Science. Proceedings of the LMS Symposium, Durham 1991*, volume 177 of *London Mathematical Society Lecture Note Series*, pages 95–106. Cambridge University Press.

Huwig, H. and Poigné, A. (1990). A note on inconsistencies caused by fixpoints in a cartesian closed category. *Theoretical Computer Science*, 73:101–112.

Jacobs, B. (1993). Comprehension categories and the semantics of type dependency. *Theoretical Computer Science*, 107:169–207.

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

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

Maietti, M. E., Maneggia, P., de Paiva, V., and Ritter, E. (2005). Relating categorical semantics for intuitionistic linear logic. *Applied Categorical Structures*, 13(1):1–36.

Maneggia, P. (2004). *Models of linear polymorphism*. PhD thesis, University of Birmingham.

Møgelberg, R. E. (2005a). *Category theoretic and domain theoretic models of parametric polymorphism*. PhD thesis, IT University of Copenhagen.

Møgelberg, R. E. (2005b). Parametric completion for models of polymorphic intuitionistic / linear lambda calculus. Technical Report TR-2005-60, IT University of Copenhagen.

Møgelberg, R. E., Birkedal, L., and Petersen, R. L. (2005). Categorical models of PILL. Technical Report TR-2005-58, IT University of Copenhagen.

O'Hearn, P. W. and Tennent, R. D. (1995). Parametricity and local variables. *Journal of the ACM*, 42(3):658–709.

Pitts, A. M. (2000). Parametric polymorphism and operational equivalence. *Mathematical Structures in computer Science*, 10:321–359.

Plotkin, G. (1993a). Second order type theory and recursion. Notes for a talk at the Scott Fest.

Plotkin, G. and Abadi, M. (1993). 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.

Plotkin, G. D. (1993b). Type theory and recursion (extended abstract). In *Proceedings, Eighth Annual IEEE Symposium on Logic in Computer Science*, page 374, Montreal, Canada. IEEE Computer Society Press.

Reynolds, J. (1983). Types, abstraction, and parametric polymorphism. *Information Processing*, 83:513–523.

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

Rosolini, G. and Simpson, A. (2004). Using synthetic domain theory to prove operational properties of a polymorphic programming language based on strictness. Manuscript.

Tse, S. and Zdancewic, S. (2004). Translating dependency into parametricity. In *ICFP '04: Proceedings of the ninth ACM SIGPLAN international conference on Functional programming*, pages 115–125. ACM Press.

A. Fibrations of opposite fibres

This appendix describes how the construction taking a fibration $p\colon \mathbb{E} \to \mathbb{B}$ producing a fibration of opposite fibres $p^{\mathrm{fop}}\colon \mathbb{E}^{\mathrm{fop}} \to \mathbb{B}^{\mathrm{fop}}$, defined in the text for split fibrations only, can be extended to general fibrations. (The construction of $\mathbb{E}^{\mathrm{fop}}$ depends not only on $\mathbb{E}$ but also on p, but we leave p implicit in notation.) Recall that since split fibrations p as above correspond to functors $\hat{p}\colon \mathbb{B}^{\mathrm{op}} \to \mathbf{Cat}$ (where $\mathbf{Cat}$ is the category of categories) via the Grothendieck construction, we could define p^{fop} in this special case to be the fibration corresponding to $(-)^{\mathrm{op}} \circ \hat{p}$, where $(-)^{\mathrm{op}}\colon \mathbf{Cat} \to \mathbf{Cat}$ is the functor mapping a category to its opposite.

Suppose now $p\colon \mathbb{E} \to \mathbb{B}$ is a fibration which is not assumed to be split. Recall that any map $f\colon A \to B$ in $\mathbb{E}$ can be written as a composition $g \circ h$ where g is cartesian and h is vertical. This composition is not unique, but for any given two such compositions $g \circ h$, $g' \circ h'$ there exists an isomorphism k such that $g' \circ k = g$ and $h' = k \circ h$. Note that k is necessarily vertical. This information tells us, that we can construct an isomorphic copy of $\mathbb{E}$ which has the same objects as $\mathbb{E}$ but with morphisms from A to B being equivalence classes of triples $(C, h\colon A \to C, g\colon C \to B)$ where h is vertical and g is cartesian, under the equivalence relation relating (C, h, g) to (C', h', g') if there exists an isomorphism $k\colon C \to C'$ such that $g' \circ k = g$ and $h' = k \circ h$.

Using this we can define $\mathbb{E}^{\mathrm{fop}}$ by just reversing the vertical arrows, defining it to be:

Objects: Objects of $\mathbb{E}$

Morphisms: A morphism from A to B is an equivalence class of triples $(C, h\colon C \to A, g\colon C \to B)$ where h is vertical and g is cartesian, under the equivalence relation relating (C, h, g) and (C', h', g') if there exists an isomorphism $k\colon C \to C'$ such that $g = g' \circ k$ and $h = h' \circ k$.

Composition in $\mathbb{E}^{\mathrm{fop}}$ is defined as follows: given a map from A to B represented by (D, h, g) and a map from B to C represented by (D', h', g') we define the composition of the two to be the map represented by $(D'', h \circ h'', g' \circ g'')$ as in the diagram

$$\begin{tikzcd}
A & & \\
D \arrow[u, "h"] \arrow[r, "g"] & B & \\
D'' \arrow[u, "h''"] \arrow[r, "g''"] & D' \arrow[u, "h'"'] \arrow[r, "g'"] & C
\end{tikzcd}$$

where g'' is a cartesian lift of $p(g)$ and h'' is the unique vertical map making the square commute. The functor $p^{\mathrm{fop}}\colon \mathbb{E}^{\mathrm{fop}} \to \mathbb{B}$ is defined to map a morphism represented by (C, h, g) to $p(g)$. The next theorem tells that all this structure is well defined. The proof is omitted.

A.1. Theorem. *The above defines a category* $\mathbb{E}^{\mathrm{fop}}$ *and a fibration* $p^{\mathrm{fop}} \colon \mathbb{E}^{\mathrm{fop}} \to \mathbb{B}$*. Moreover,* $(-)^{\mathrm{fop}}$ *extends to an endofuntor on the category* $\mathbf{Fib}(\mathbb{B})$ *of fibrations over* $\mathbb{B}$*.*

Lars Birkedal and Rasmus E. Møgelberg
IT University of Copenhagen
Rued Langgaardsvej 7
DK–2300 Copenhagen S
Denmark

Rasmus L. Petersen
Department of Computer Science
Queen Mary, University of London
London E1 4NS
UK

This article may be accessed at `http://www.tac.mta.ca/tac/` or by anonymous ftp at `ftp://ftp.tac.mta.ca/pub/tac/html/volumes/20/7/20-07.{dvi,ps,pdf}`