# Parametric limits

Dunphy, BP; Reddy, Uday

*DOI:*
10.1109/LICS.2004.1319618

# Parametric Limits

Brian Dunphy
University of Illinois
dunphy@math.uiuc.edu

Uday S. Reddy
University of Birmingham
u.reddy@cs.bham.ac.uk

## Abstract

*We develop a categorical model of polymorphic lambda calculi using a notion called parametric limits, which extend the notion of limits in categories to reflexive graphs of categories. We show that a number of parametric models of polymorphism can be captured in this way. We also axiomatize the structure of reflexive graphs needed for modelling parametric polymorphism based on ideas of fibrations, and show that it leads to proofs of representation results such as the initial algebra and final coalgebra properties one expects in polymorphic lambda calculi.*

**Keywords:** *Parametric polymorphism, Relational Parametricity, Categorical models, Fibrations.*

## 1 Introduction

In his seminar paper connecting polymorphism and type abstraction [26], Reynolds proposed that a polymorphic type $\forall X.\, F(X)$ should not be interpreted simply as a product $\prod_{X \in \mathcal{S}} F(X)$ consisting of all $\mathcal{S}$-indexed families but rather as the set of "parametric families." In Reynolds's formulation an $\mathcal{S}$-indexed family $\langle p_X \rangle_X \in \prod_{X \in \mathcal{S}} F(X)$ is parametric if:

> for all relations $R\colon X \leftrightarrow X'$, the components of the family $p_X$ and $p_{X'}$ are related by $F(R)$.

Reynolds reasoned that the computations expressible in typical polymorphic type systems naturally satisfy the parametricity property.

In category theory, we find a closely related notion, viz., that of *limits*. If $F\colon \mathbf{D} \to \mathbf{Set}$ is a functor, the limit $\mathrm{Lim}_X F(X)$ is the set of $\mathbf{D}$-indexed families $\langle p_X \rangle_X \in \prod_{X \in \mathrm{Ob}(\mathbf{D})} F(X)$ that satisfy the condition:

> for all morphisms $f\colon X \to X'$ in $\mathbf{D}$, the components $p_X$ and $p_{X'}$ are related by $p_{X'} = F(f)(p_X)$.

Of course, the categorical notion is not limited to sets. The limit of a functor $F\colon \mathbf{D} \to \mathbf{C}$ is defined by a suitable universal property and works for any category $\mathbf{C}$. If we could adapt the notion of limits to cover Reynolds's idea, then we would have achieved the lifting of parametricty to a categorical level so that it applies to a wide range of contexts instead of just sets.

In this paper, we accomplish this task. Categories by themselves have objects and morphisms but nothing resembling relations. We follow the suggestion of O'Hearn and Tennent [17] (see also [29]) that reflexive graphs of categories can be used to model "categories with relations." It turns out that reflexive graphs of categories have pleasant properties that make them almost like categories. Once we work through these properties, there is a natural notion of limits in reflexive graphs which is what we are after. By suitably reformulating the notion of limits as what we call "parametric limits," we obtain a categorical generalization of Reynolds's parametricity. Using it, we are able to define categorical models for predicative polymorphic lambda calculus, and, by internalizing the definitions, for the (impredicative) polymorphic lambda calculus as well.

Having defined a categorical model, one should ask how good the model is. For polymorphic lambda calculus, there is general agreement that a model is parametric if it validates isomorphisms representing initial algebras and final coalgebras [2, 8, 5, 26, 21, 32]. The general model of the above kind has no reason to be parametric. In fact, an arbitrary reflexive graph has no reason to have any nontrivial relations and, in such a situation, a parametric limit would be nothing but an ordinary product. To obtain a parametric model, we need to ensure that there are "enough relations." This we do by borrowing a chapter from the theory of fibrations. We produce an axiomatization of the structure necessary for reflexive graphs to obtain parametric models, and show that initial algebra/final coalgebra results follow.

Our work is a modest contribution to category theory (after all, the intuitive notion of uniformity was the driving force in the development of category theory) and perhaps a significant contribution to parametricty theory. Reynolds's formulation of parametricity, being limited to sets, suffers some strain in its application to a wide range of contexts where parametricity notions are needed, especially in the semantics of stateful programming languages [17, 16, 22, 23, 24]. By reformulating polymorphic type quantification as a natural form of limits, characterized by a universal property, we are able to offer a clean definition that is applicable in a wide range of contexts. Most of all, we are pleased to be in a position to say that Reynolds's definition is an inevitable one, once its setting is recognized.

### 1.1 Related work

Mitchell and Scedrov [15] made the first attempt at categorical modelling of a lambda calculus with implicit polymorphism. Their structure amounts to using a graph category. This is too little structure to model polymorphic lambda calculus, but some of the first steps in incorporating relations in a categorical setting have been taken here. O'Hearn and Tennent [17] promoted the structure to reflexive graph categories and used it to analyze the parametricity properties of Algol-like languages. They made no attempt to model polymorphic lambda calculus.

Ma and Reynolds [13] gave the first categorical model of parametricity in the framework of PL-categories [31]. Their structure amounts to a reflexive graph of PL-categories with a certain parametricity condition which applies to the interpretations of closed types. Using this condition, it is possible to show isomorphisms involving closed types, but nothing can be said about types with free variables. The categorical model we describe is a parametric model in the sense of Ma and Reynolds.

Bainbridge et al. [2] gave the first concrete parametric model of polymorphic lambda calculus by trimming down the PER model using relations. Robinson and Rosolini [29] describe a class of models built using jointly monic spans. All these models are special cases of the categorical model we describe.

Representation results of the kind we prove were first discovered by Reynolds [26] and were later improved in [28] and [19]. Our proofs show that they hold in a large class of models, not only the the specific models considered previously.

## 2 Background and definitions

We first recall the definition of limits in categories. If $F\colon \mathbf{D} \to \mathbf{C}$ is a functor, the *limit* of $F$, denoted $\mathrm{Lim}(F)$ or $\mathrm{Lim}_X F(X)$, is an object of $\mathbf{C}$ together with a natural transformation $\omega\colon \mathrm{Lim}_X F(X) \to F$ that is universal, i.e., any other natural transformation $\beta\colon A \to F$ from an object $A$ of $\mathbf{C}$ uniquely factors through $\omega$.

$$\begin{tikzcd} \mathrm{Lim}_X F(X) \arrow[r, "\omega"] & F \\ A \arrow[u, dashed, "\beta^*"] \arrow[ur, "\beta"'] & \end{tikzcd}$$

(Recall that, in writing an object $A$ as a functor, we mean the constant functor $\Delta_A\colon \mathbf{D} \to \mathbf{C}$ that sends every object to $A$ and every morphism to $\mathrm{id}_A$.) We call this the "pointwise" definition of limits. If every functor $F\colon \mathbf{D} \to \mathbf{C}$ has a limit, then these limits can be gathered together into a functor $\mathrm{Lim} : \mathbf{C}^{\mathbf{D}} \to \mathbf{C}$. The limit functor can be defined in a more straightforward fashion: *it is the right adjoint to the diagonal functor* $\Delta\colon \mathbf{C} \to \mathbf{C}^{\mathbf{D}}$. The pointwise definition is merely an unfolding of this general notion.

The relationship of limits to Reynolds's parametricity can be seen by considering the case where $\mathbf{C} = \mathbf{Set}$. In that case, $\mathrm{Lim}_X F(X)$ can be characterized as in the Introduction, the set of $\mathbf{D}$-indexed families $\langle p_X \rangle_X$ such that, for all $f\colon X \to X'$ in $\mathbf{D}$, $F(f)$ sends $p_X$ to $p_{X'}$. The natural transformation $\omega$ picks out the components of the family and, for any natural transformation $\beta\colon A \to F$, $\beta^*$ maps $a \in A$ to the family $\langle \beta_X(a) \rangle_X$. *Naturality of $\beta$ guarantees that this family is an element of the limit.* In effect, the notion of limit internalizes the concept of natural transformation.

### Reflexive graphs

Since categories do not have "relations" required for parametricity, we model them in a separate category whose objects are abstractly called "edges." (The base category is then called the vertex category.) The morphisms in the edge category model "relation-preserving functions." Two functors $\partial_0$ and $\partial_1$ from the edge category to the vertex category pick out the source and target of edges and edge morphisms, and a functor $I$ from the vertex category to the edge category picks out the "identity edge:"

**Definition 1** A *reflexive graph category* $\mathbf{G}$ consists of

two categories and three functors as depicted in:

$$\begin{tikzcd} \mathbf{G}_v \arrow[r, "I", shift left=2] \arrow[r, shift right=2, "\partial_0"'] \arrow[r, shift right=4, "\partial_1"'] & \mathbf{G}_e \end{tikzcd}$$

such that $I;\partial_0 = \mathrm{id}_{\mathbf{G}_v} = I;\partial_1$.

**Notation** We use the terminology "vertex" and "arrow" for the objects and morphisms of $\mathbf{G}_v$, and "edge" and "square" for those of $\mathbf{G}_e$. The identity edge $I(A)$ is written as $I_A$. If $R$ is an edge such that $\partial_0 R = A_0$ and $\partial_1 R = A_1$ then we write $R\colon A_0 \leftrightarrow A_1$. Similarly for edge morphisms $\psi\colon f_0 \leftrightarrow f_1$. However, since edge morphisms also have sources and targets in the sense of morphisms $\psi\colon R \to S$, all the common vertices have to match, as in the two-dimensional diagram:

$$\begin{tikzcd} A_0 \arrow[r, "f_0"] \arrow[d, leftrightarrow, "R"'] \arrow[dr, phantom, "\psi"] & B_0 \arrow[d, leftrightarrow, "S"] \\ A_1 \arrow[r, "f_1"] & B_1 \end{tikzcd}$$

The diagram, consisting of all the typing data of the square $\psi$, is called the *shape* of $\psi$.

If a reflexive graph is such that there is at most one square of a given shape, it is called a *relational* reflexive graph. (Then a square $R \to S$ merely determines a relation between hom-sets $\mathrm{Hom}[A_0, B_0]$ and $\mathrm{Hom}[A_1, B_1]$.)

The canonical example of reflexive graphs is **REL**, whose vertex category is **Set** and the edge category has relations $\langle A_0, A_1, R \subseteq A_0 \times A_1\rangle$ as objects, and pairs of functions $\langle f_0, f_1\rangle : R \to S$ as morphisms whenever $f_0\ [R \to S]\ f_1$. The functors $\partial_i$ map $\langle A_0, A_1, R\rangle$ to $A_i$ and $\langle f_0, f_1\rangle$ to $f_i$. The $I$ functor sends $A$ to the diagonal relation $\langle A, A, \Delta_A\rangle$ and $f$ to $\langle f, f\rangle$. note that this is *relational* reflexive graph.

Familiar categories such as **Poset**, **Cpo**, $\mathbf{Cpo}_\perp$, **Cppo** etc. can be similarly given a reflexive graph structure by picking suitable forms of relations as edges.

Any category $\mathbf{C}$ gives rise to three immediate reflexive graphs. The category $\mathbf{C}$ itself can be treated as a reflexive graph by taking $\mathbf{C}$ as both the vertex and edge category with $I_A = A$. (There are no non-identity edges.) The reflexive graph $\mathbf{C}^{\to}$ uses the arrow category of $\mathbf{C}$ as the edge category with $I_A = \mathrm{id}_A$. Finally, $\mathbf{Span}(\mathbf{C})$ has spans $A_0 \xleftarrow{r_0} R_w \xrightarrow{r_1} A_1$ in $\mathbf{C}$ as edges and span-morphisms as edge morphisms.

$$\begin{tikzcd} & A_0 \arrow[rr, "f_0"] & & B_0 \\ R_w \arrow[ur] \arrow[rr, "f_w"] \arrow[dr] & & S_w \arrow[ur] \arrow[dr] & \\ & B_0 \arrow[rr, "f_1"] & & B_1 \end{tikzcd}$$

Note that $\mathbf{Span}(\mathbf{C})$ is not a relational reflexive graph. Instead of all spans, one can consider jointly monic spans [29], which gives rise to a reflexive graph $\mathbf{JMS}(\mathbf{C})$. It shares most formal properties of $\mathbf{Span}(\mathbf{C})$ but it is relational.

## The 2-category RG

Reflexive graph categories can be reduced to the familiar setting of indexed categories. Let $\mathbf{B}$ denote the category with two objects $v$ and $e$ and five non-identity morphisms as follows:

$$\begin{tikzcd} v \arrow[r, shift left=2, "I"] \arrow[r, shift right=2, "\partial_0"'] \arrow[r, shift right=4, "\partial_1"'] & e \arrow[r, shift left=2, "\partial_0;I"] \arrow[r, shift right=2, "\partial_1;I"'] & e \end{tikzcd}$$

Then a reflexive graph category $\mathbf{G}$ is nothing but a functor $\mathbf{G}\colon \mathbf{B} \to \mathbf{CAT}$, in other words a $\mathbf{B}^{\mathrm{op}}$-indexed category.

We have a 2-category $\mathbf{RG}$ with the data:

| | |
|---|---|
| objects: | Reflexive graph categories |
| | $= \mathbf{B}^{\mathrm{op}}$-indexed categories |
| 1-cells: | $\mathbf{B}^{\mathrm{op}}$-indexed functors |
| 2-cells: | $\mathbf{B}^{\mathrm{op}}$-indexed natural transformations |

(We suppress mentioning $\mathbf{B}^{\mathrm{op}}$ in the sequel.) Explicating the concepts, an indexed functor $F\colon \mathbf{G} \to \mathbf{H}$ is a pair of functors $\langle F_v\colon \mathbf{G}_v \to \mathbf{H}_v,\ F_e\colon \mathbf{G}_e \to \mathbf{H}_e\rangle$ that commutes with the structural functors of reflexive graphs. An indexed natural transformation $\eta\colon F \to G : \mathbf{G} \to \mathbf{H}$ is similarly a pair $\langle \eta_v\colon F_v \to G_v,\ \eta_e\colon F_e \to G_e\rangle$ of natural transformations that preserve the structural functors: $\partial_i\eta_e = \eta_v\partial_i$ and $I\eta_v = \eta_e I$. (We omit the subscripts $v$ and $e$ from the components of $F$ and $\eta$ whenever they clear from the context.) Thus, an indexed natural transformation $\eta$ has first an ordinary natural transformation $\eta_v$ and, second, for each edge $R\colon A \leftrightarrow A'$ of $\mathbf{G}$, a square of the form:

$$\begin{tikzcd} FA \arrow[r, "\eta_A"] \arrow[d, leftrightarrow, "FR"'] \arrow[dr, phantom, "\eta_R"] & GA \arrow[d, leftrightarrow, "GR"] \\ FA' \arrow[r, "\eta_{A'}"] & GA' \end{tikzcd}$$

For *relational* reflexive graphs, $\eta_R$ is unique whenever it exists. So, its existence is all that matters.

Since **RG** is a 2-category, all the usual category theory of adjunctions, monads, algebras etc. carries over to them. For instance, "cartesian closed reflexive graph category" is a well-defined concept since products and exponents are adjoints to suitable indexed functors.

To get some sense of the broader applicability of the ideas, consider the case of $F$-algebras, where $F\colon \mathbf{REL} \to \mathbf{REL}$ is an indexed functor. As usual, an $F$-algebra is a pair $\langle A,\ \alpha\colon FA \to A\rangle$ and morphisms are algebra-morphisms. For edges, the natural candidates are $F$-simulations, i.e., relations $R\colon A \leftrightarrow A'$ such that $\alpha\ [FR \to R]\ \alpha'$. (See, for instance, [30].) We thus obtain a reflexive graph $F$-**Alg**.

## 3 Parametric limits

Whenever **C** and **D** are categories, there is a functor category $\mathbf{C}^{\mathbf{D}}$ whose objects are functors $\mathbf{D} \to \mathbf{C}$ and morphisms are natural transformations. This feature does not carry over to indexed categories in general. In fact, indexed categories may not have anything resembling "objects." However, for reflexive graphs, this feature does carry over and motivates our observation that reflexive graph categories are very much "like categories."

**Theorem 2** Reflexive graph categories have functor categories, i.e., the exponent $\mathbf{H}^{\mathbf{G}}$ of two reflexive graphs **G** and **H** is a reflexive graph whose vertices are indexed functor $\mathbf{G} \to \mathbf{H}$ and arrows are indexed natural transformations.

*Proof*: Indexed categories are cartesian closed, and their exponents are given in the same way as in presheaf-categories, using a hom-functor. The hom-functor in the case of reflexive graphs is of the form $\mathrm{Hom}_{\mathbf{B}}[w,-] : \mathbf{B} \to \mathbf{CAT}$ where $w$ is an object of **B**. Calculation shows that we have discrete reflexive graph categories:

| | | | |
|---|---|---|---|
| $\mathrm{Hom}_{\mathbf{B}}[v,-]$ | $\cong$ | $\mathbf{1}$ | (one vertex, only identity edge) |
| $\mathrm{Hom}_{B}[e,-]$ | $\cong$ | $\mathbf{E}$ | (two vertices, one non-identity edge) |

It follows that $\mathbf{H}^{\mathbf{G}}$ has:

| | |
|---|---|
| vertex category: | indexed functors $\mathbf{G} \to \mathbf{H}$ and indexed natural transformations |
| edge category: | indexed functors $\mathbf{E}\times\mathbf{G} \to \mathbf{H}$ and indexed natural transformations |

In other words, an edge $\rho\colon F \to F'$ in $\mathbf{H}^{\mathbf{G}}$ is given by a family $\langle \rho_R\rangle_R$, indexed by edges $R\colon X \leftrightarrow X'$ in **G**, of edges $\rho_R\colon FX \leftrightarrow F'X'$ in **H**. a square of shape on the left, below, consists of $R$-indexed families of squares of shape on the right:

$$\begin{tikzcd} F \arrow[r, "\eta"] \arrow[d, leftrightarrow, "\rho"'] & G \arrow[d, leftrightarrow, "\sigma"] \\ F' \arrow[r, "\eta'"] & G' \end{tikzcd} \qquad \begin{tikzcd} FX \arrow[r, "\eta_X"] \arrow[d, leftrightarrow, "\rho_R"'] & GX \arrow[d, leftrightarrow, "\sigma_R"] \\ F'X' \arrow[r, "\eta'_{X'}"] & G'X' \end{tikzcd}$$

■

For any pair of reflexive graphs **G** and **H**, we have a diagonal indexed functor $\Delta\colon \mathbf{H} \to \mathbf{H}^{\mathbf{G}}$ whose action is described as follows:

- For a vertex $A$, $\Delta(A)$ is the constant indexed functor that sends every vertex of **G** to $A$, every morphism to $\mathrm{id}_A$, every edge to $I_A$ and every square to $I_{\mathrm{id}_A}$.
- For an edge $R\colon A \leftrightarrow A'$, $\Delta(R)$ is the constant family that maps every edge of **G** to $R$, and every square to $\mathrm{id}_R$.

(Note the crucial role played by the identity edge $I_A$ in this definition.)

We now define

| | | |
|---|---|---|
| parametric limits | $\forall : \mathbf{H}^{\mathbf{G}} \to \mathbf{H}$ | right adjoint to $\Delta$ |
| parametric colimits | $\exists : \mathbf{H}^{\mathbf{G}} \to \mathbf{H}$ | left adjoint to $\Delta$ |

Unpacking the definition, we explicate what happens with pointwise parametric limits. If $F\colon \mathbf{G} \to \mathbf{H}$ is an indexed functor, the parametric limit of $F$ denoted $\forall_X F(X)$ is a vertex of **H** together with an indexed natural transformation $\omega\colon \forall_X F(X) \to F$ that is universal, i.e., any other indexed natural transformation $\beta\colon A \to F$ from any vertex $A$ of **H** uniquely factors through $\omega$:

$$\begin{tikzcd} \forall_X F(X) \arrow[r, "\omega"] & F \\ A \arrow[u, dashed, "\beta^*"] \arrow[ur, "\beta"'] & \end{tikzcd}$$

In treating a vertex $A$ as an indexed functor, we mean the constant functor $\Delta(A)\colon \mathbf{G} \to \mathbf{H}$. Here are some examples.

The reflexive graph **REL** has small parametric limits and colimits.

- If $F\colon \mathbf{G} \to \mathbf{REL}$ is an indexed functor (and **G** is small), then

$$\begin{aligned} \forall_X F(X) = \ \{ \langle t_X\rangle_X \in \textstyle\prod_{X\in \mathrm{Ob}(\mathbf{G}_v)} F(X) \mid & \\ \forall R\colon X \leftrightarrow X'.\ t_X\ [F(R)]\ t_{X'} \wedge & \\ \forall f\colon X \to X'.\ F(f)(t_X) = t_{X'}\ \} & \end{aligned}$$

Given any indexed natural transformation $\beta: A \to F$, $\beta^* : A \to \forall_X F(X)$ is defined by $\beta^*(a) = \langle \beta_X(a) \rangle_{X \in \mathrm{Ob}(\mathbf{G}_v)}$. The fact that $\beta$ is an indexed natural transformation guarantees that $\beta^*(a) \in \forall_X F(X)$.

- The parametric colimit $\exists_X F(X)$ is the quotient of $\sum_{X \in \mathrm{Ob}(\mathbf{G}_v)} F(X)$ with respect to the least equivalence relation generated by the following conditions:
  - $R: X \leftrightarrow X' \wedge t\ [F(R)]\ t' \Longrightarrow \langle X, t \rangle \equiv \langle X', t' \rangle$, and
  - $f: X \to X' \Longrightarrow \langle X, t \rangle \equiv \langle X', f(t) \rangle$.

The constructions extend to indexed functors $\mathbf{REL}^{\mathbf{G}} \to \mathbf{REL}$. If $\rho: F \leftrightarrow F'$ is an edge in $\mathbf{REL}^{\mathbf{G}}$ then $\forall(\rho)$ and $\exists(\rho)$ are relations given by:

$$\langle p_X \rangle_X\ [\forall(\rho)]\ \langle p'_X \rangle_X \iff \forall R: X \leftrightarrow X'.\, p_X\ [\rho_R]\ p'_{X'}$$

$$[\langle X, p \rangle]\ [\exists(\rho)]\ [\langle X', p' \rangle] \iff \exists R: Y \leftrightarrow Y'.\, \exists \langle Y, q \rangle \in [\langle X, p \rangle].\, \exists \langle Y', q' \rangle \in [\langle X', p' \rangle].\ q\ [\rho_R]\ q'$$

Note that these constructions coincide with those of Reynolds [26] for the universal quantification and Reddy [23] for the existential quantification (with the proviso that they do not include the naturality conditions because their reflexive graphs $\mathbf{G}$ are discrete.)

**CPO** has small parametric limits and colimits given in a similar fashion. See [4] for details.

**Theorem 3** If $\mathbf{C}$ is a category with small limits then $\mathbf{Span}(\mathbf{C})$ has all small parametric limits. If $\mathbf{C}$ has small colimits then $\mathbf{Span}(\mathbf{C})$ has small parametric colimits.

*Proof sketch*: If $F: \mathbf{G} \to \mathbf{Span}(\mathbf{C})$ is an indexed functor, we construct a category $\mathbf{G}^*$ whose objects are all the vertices and edges of $\mathbf{G}$. The arrows of $\mathbf{G}^*$ are pairs $(m, f): X \to Y$ where $m: x \to y$ is an arrow of $\mathbf{B}$ and $f: \mathbf{G}(m)(X) \to Y$ is an arrow of $\mathbf{G}_y$. The indexed functor $F$ determines a functor $F^* : \mathbf{G}^* \to \mathbf{C}$ such that $\forall_X F(X) = \mathrm{Lim}_X F^*(X)$ and $\exists_X F(X) = \mathrm{Colim}_X F^*(X)$. ■

## 4 Modelling polymorphic lambda calculi

Relational parametricity arose from Reynolds's attempt to give a (classical) model of the impredicative polymorphic lambda calculus or System F [6, 25]. Even though this attempt was later shown to be infeasible [27, 28], it can be made to succeed for a predicative calculus representing the type systems in languages like ML and Haskell [14]. The same ideas can be recast in an internal category setting to model the impredicative calculus as well. In this section, we define categorical models of both the cases using parametric limits.

A predicative polymorphic lambda calculus has a type structure of the following form:

$$\begin{array}{lcll} \text{simple types} & \tau & ::= & \alpha \mid \tau_1 \times \tau_2 \mid \tau_1 \to \tau_2 \\ \text{polytypes} & \sigma & ::= & \tau \mid \forall \alpha.\, \sigma \mid \exists \alpha.\, \sigma \end{array}$$

where $\alpha$ ranges over type variables. (We leave open the possibility of adding primitive types.)

A categorical model for the predicative calculus consists of:

- a reflexive graph category $\mathbf{G}$, and
- a small cartesian closed reflexive graph subcategory $\mathbf{S}$ of $\mathbf{G}$ (with an inclusion functor $J : \mathbf{S} \to \mathbf{G}$)

such that there are parametric limit and parametric colimit functors $\forall, \exists : \mathbf{G}^{|\mathbf{S}|} \to \mathbf{G}$, where $|\mathbf{S}|$ denotes the discrete reflexive graph of $\mathbf{S}$ (obtained by dropping all non-identity morphisms).

A simple type expression with $n$ type variables is interpreted as an indexed functor $|\mathbf{S}|^n \to \mathbf{S}$:

$$\begin{array}{rcl} [\![\alpha_i]\!] & = & \Pi_i \qquad \text{(the } i\text{th projection)} \\ [\![\tau_1 \times \tau_2]\!] & = & \times \circ \langle [\![\tau_1]\!], [\![\tau_2]\!] \rangle \\ [\![\tau_1 \to \tau_2]\!] & = & \Rightarrow \circ \langle [\![\tau_1]\!], [\![\tau_2]\!] \rangle \end{array}$$

A polytype expression with $n$ type variables is interpreted as an indexed functor $|\mathbf{S}|^n \to \mathbf{G}$:

$$\begin{array}{rcl} [\![\tau]\!] & = & J \circ [\![\tau]\!] \\ [\![\forall \alpha_{n+1}.\, \sigma]\!] & = & \forall^{|\mathbf{S}|^n}([\![\sigma]\!]) \\ [\![\exists \alpha_{n+1}.\, \sigma]\!] & = & \exists^{|\mathbf{S}|^n}([\![\sigma]\!]) \end{array}$$

where $\forall^{|\mathbf{S}|^n}, \exists^{|\mathbf{S}|^n} : (\mathbf{G}^{|\mathbf{S}|^n})^{|\mathbf{S}|} \to \mathbf{G}^{|\mathbf{S}|^n}$ are the parametric limit and colimit functors. The interpretation of terms can be given in the evident fashion using the combinators associated with parametric limits and colimits.

The impredicative polymorphic lambda calculus has the type structure:

$$\sigma ::= \alpha \mid \sigma_1 \to \sigma_2 \mid \forall \alpha.\, \sigma$$

A categorical model for this calculus consists of a cartesian-closed reflexive graph category $\mathbf{G}$ such that the parametric limit indexed functor $\forall: \mathbf{G}^{|\mathbf{G}|} \to \mathbf{G}$ exists. There are no reflexive graphs of this kind internal to **Set**. Reflexive graphs, like **PER**, that are internal to $\omega - \mathbf{Set}$ [12] or, alternatively, the effective

topos [10], satisfy the condition. (We omit the details of restating the definitions to an internal category setting, which may be found in [4].)

An important point to note is that the type expressions are interpreted as functors $|\mathbf{S}|^n \to \mathbf{G}$ or $|\mathbf{G}|^n \to \mathbf{G}$ from a *discrete* reflexive graph. (We cannot use the category $\mathbf{S}^n$ directly because type variables may be used in both covariant and contravariant positions and type expressions would not be functorial in the type variables.) Thus, the only uniformity conditions in the interpretation of terms are those arising from the preservation of edges. If we had used plain categories instead of reflexive graph categories, no uniformity conditions would be captured and the interpretation of $\forall$ would amount to a simple product.

## 5 Axiomatizing parametricity

Reflexive graphs add structure to categories so that we can incorporate Reynolds's ideas of relation-preservation. However, they do not ensure that relation-preservation has in fact been captured. Since any category can be trivially regarded as a reflexive graph (Sec. 2), the setting of reflexive graphs by itself does not impose any new conditions on the models. Specifically, we can identify three issues:

- There may not be enough relations to ensure that relation-preservation is a meaningful constraint.
- The identity edges need to be given semantics (just as identity morphism acquire semantics via the condition that they be the units of composition).
- Multiple squares can be present witnessing the fact that a pair of arrows bear a particular relation.

We address these issues in the sequel.

### 5.1 Relational reflexive graphs

In Reynolds's model of the polymorphic lambda calculus, relations were introduced as a tool to describe the uniformity properties of collections of functions. However, in our categorical model, terms are interpreted as indexed natural transformations – which are collections of arrows and squares, where the latter denote morphisms between relations. We cannot suppress the information about squares because it matters which squares are used in a collection.

Consider the interpretation of $\forall X.\, X \to X$ in $\mathbf{Span}(\mathbf{Set})$. The interpretation consists of indexed natural transformations of type $1 \to (J \Rightarrow J)$ or, equivalently, those of type $J \to J$. Other than the identity transformation, there are countless other transformations of type $J \to J$, which differ in the witnesses used in the spans. For instance, let $R_w = \{a, b\}$ in the span $1 \longleftarrow R_w \longrightarrow 1$. If $m$ is a permutation of $R_w$ then there is an indexed natural transformation $\tau\colon J \to J$ which acts as the permutation $m$ for the span $R$ and as identity for every other span. Thus, the intuitive idea that $\forall X.\, X \to X$ consists of a single polymorphic function is lost.

We have introduced *relational* reflexive graphs as those that have at most one square of any given shape. Relational reflexive graphs do not have this problem. There is a 2-category of relational reflexive graphs $\mathbf{rRG}$ which is a 2-subcategory of $\mathbf{RG}$.

If one wants to consider non-relational reflexive graphs such as spans, then there is an easy device to convert them into relational reflexive graphs by collapsing all the squares of a particular shape into a single equivalence class. This operation determines a 2-functor $\mathcal{R}\colon RG \to \mathbf{rRG}$, which is left adjoint to the inclusion $\mathbf{rRG} \to \mathbf{RG}$. In other words, $\mathbf{rRG}$ is a reflective 2-subcategory of $\mathbf{RG}$.

### 5.2 Fibred reflexive graphs

Hermida [9] proposed that logical relations are best modelled in terms of fibrations. Since fibrations have a strong connection with predicate logic, this is a fruitful way to capture the idea that edges model relations.

A reflexive graph $\mathbf{G}$ is said to be *fibred* if $\langle \partial_0, \partial_1 \rangle : \mathbf{G}_e \to \mathbf{G}_v \times \mathbf{G}_v$ is a fibration. More explicitly, we give the following definition:

**Definition 4** a square
$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, leftrightarrow, "P"'] \arrow[dr, phantom, "\phi"] & B \arrow[d, leftrightarrow, "R"] \\ A' \arrow[r, "f'"] & B' \end{tikzcd}$$
is said to be *cartesian* if for any square $\psi$ of the shape on the left, below, there is a unique factorization of $\psi$ through $\phi$. We denote the factor by $\Phi(\psi)$.

$$\begin{tikzcd} X \arrow[r, "f \circ g"] \arrow[d, leftrightarrow, "T"'] \arrow[dr, phantom, "\psi"] & B \arrow[d, leftrightarrow, "R"] \\ X' \arrow[r, "f' \circ g'"] & B' \end{tikzcd} \qquad \begin{tikzcd} X \arrow[r, "g"] \arrow[d, leftrightarrow, "T"'] \arrow[dr, phantom, "\Phi(\psi)"] & A \arrow[r, "f"] \arrow[d, leftrightarrow, "P"] \arrow[dr, phantom, "\phi"] & B \arrow[d, leftrightarrow, "R"] \\ X' \arrow[r, "g'"] & A' \arrow[r, "f'"] & B' \end{tikzcd}$$

A reflexive graph is said to be *fibred* if, for every edge $R\colon B \leftrightarrow B'$ and morphisms $f\colon A \to B$, $f'\colon A' \to B'$, there exists an edge $[f, f']R : A \leftrightarrow A'$ with a cartesian

edge morphism of the following shape:

$$\begin{tikzcd} A \arrow[r, "f"] & B \\ A' \arrow[u, leftrightarrow, "{[f,f']R}"'] \arrow[r, "f'"] & B' \arrow[u, leftrightarrow, "R"'] \end{tikzcd}$$

Such an edge is said to be the *weakest pre-edge* of $R$ along $(f, f')$.

The notion of weakest pre-edge has obvious parallels with the predicate transformer semantics used in imperative programming [3, 20] as well as modalities in dynamic logic [7].

The reflexive graph **REL** is fibred. The weakest pre-edge $[f, f']R$ is given by:

$$x\ [[f, f']R]\ x' \iff f(x)\ [R]\ f'(x')$$

The reflexive graph **CPO** is fibred in the same way. If **C** is a category with all finite limits, **Span**(**C**) is fibred, with the weakest pre-edge $[f_0, f_1]R$ given by the limit of the diagram:

$$\begin{tikzcd} A_0 \arrow[rr, "f_0"] & & B_0 \\ & R_w \arrow[ur] \arrow[dr] & \\ A_1 \arrow[rr, "f_1"] & & B_1 \end{tikzcd}$$

In contrast, note that the trivial examples **C** and $\mathbf{C}^{\to}$ are not fibred in general.

Weakest pre-edges are unique up to isomorphism. A specific choice of weakest pre-edges is called a cleavage. We assume that a specific cleavage is given in specifying a fibred reflexive graph. Further, we assume that the cleavage is chosen so that $[\mathrm{id}_X, \mathrm{id}_{X'}]R = R$ (called a "normalized" cleavage.)

**Definition 5** If **G** and **H** are fibred reflexive graphs with chosen cleavages, a *cloven indexed functor* $F\colon \mathbf{G} \to \mathbf{H}$ is one that preserves the chosen cleavage: $F([f, f']R) = [Ff, Ff']FR$.

### 5.3 Identity condition

To give semantics to the identity edges, we borrow the following notion from Kinoshita et al. [11]:

**Definition 6** A reflexive graph category **G** is said to satisfy the *identity condition* if, whenever there is a square of shape

$$\begin{tikzcd} A \arrow[r, "f"] & B \\ A' \arrow[u, leftrightarrow, "I_A"'] \arrow[r, "g"] & B' \arrow[u, leftrightarrow, "I_B"'] \end{tikzcd}$$

, it follows that $f = g$.

Note that the converse of this condition is automatic because, whenever $f = g$, $I_f$ is a square of the required shape.

Fibred reflexive graphs with identity condition are *subsumptive* [22] in the sense that all commuting squares of the vertex category have squares witnessing them. Define a mapping of arrows $f\colon X \to X'$ to edges $\langle f\rangle\colon X \leftrightarrow X'$ in a fibred reflexive graph by $\langle f\rangle = [f, \mathrm{id}_{X'}]I_{X'}$. For example, in **REL**, $\langle f\rangle$ denotes the graph of the function $f$:

$$\langle f\rangle = \{\, (x, x') \mid f(x) = x' \,\}$$

**Theorem 7** In a fibred reflexive graph **G** satisfying the identity condition,

- for every vertex $A$, $\langle \mathrm{id}_A\rangle = I_A$, and
- there is a square of shape on the left, below, if and only if the diagram on the right commutes:

$$\begin{tikzcd} A \arrow[r, "f"] & B \\ A' \arrow[u, leftrightarrow, "\langle g\rangle"'] \arrow[r, "f'"] & B' \arrow[u, leftrightarrow, "\langle h\rangle"'] \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "h"] \\ A' \arrow[r, "f'"] & B' \end{tikzcd}$$

The first significant result following from our axiomatization is the following:

**Theorem 8** In a fibred reflexive graph **G** with the identity condition, the parametric limit $\forall_X X$ and the initial vertex of **G** are the same concept (if one exists, so does the other and it is identical.)

A cloven indexed functor $F\colon \mathbf{G} \to \mathbf{H}$ evidently preserves the subsumption map: $F(\langle f\rangle) = \langle Ff\rangle$. This fact is useful when we consider the modelling of a polymorphic lambda calculus. As remarked in Sec. 4, the variance issues force the model to ignore the action of type constructors on morphisms. So, the uniformity conditions represented by naturality are not captured. However, the subsumption of morphisms by edges means that nothing is lost.

**Definition 9** Let $F, G : \mathbf{G} \to \mathbf{H}$ be indexed functors. A family of arrows $\{\tau_X\colon FX \to GX\}_{X \in \mathrm{Ob}(\mathbf{G}_v)}$ is called a *parametric transformation* if, for every edge $R\colon X \leftrightarrow X'$ of **G**, there is a square of the following shape:

$$\begin{tikzcd} FX \arrow[r, "\tau_X"] & GX \\ FX' \arrow[u, leftrightarrow, "FR"'] \arrow[r, "\tau_{X'}"] & GX' \arrow[u, leftrightarrow, "GR"'] \end{tikzcd}$$

**Theorem 10** Let $F, G\colon \mathbf{G} \to \mathbf{H}$ be cloven indexed functors. Then any parametric transformation $\tau\colon F \to G$ is a natural transformation $F_v \to G_v$.

Thus the point of our axiomatization is that parametricity subsumes the naturality condition. Such subsumption works even in the presence of mixed variance.

**Theorem 11** Let $F, G : \mathbf{G}^{\mathrm{op}} \times \mathbf{G} \to \mathbf{H}$ be cloven indexed functors. Then any parametric transformation $\tau\colon F\Delta \to G\Delta$ is a dinatural transformation $F_v \to G_v$.

### 5.4 Parametricity graphs

Gathering all the axioms together, we coin a new term:

**Definition 12** A *parametricity graph* is a fibred, relational reflexive graph (with a chosen normalized cleavage) satisfying the identity condition.

A *PG-functor* is a cloven indexed functor between parametricity graphs.

The 2-category **PG** is made up of parametricity graphs, PG-functors and parametric transformations.

Examples of parametricity graphs include **REL**, **CPO**, **PER**, $F$-**Alg** (provided $F$ is fibred), $\mathcal{R}(\mathbf{Span}(\mathbf{C}))$ and $\mathbf{JMS}(\mathbf{C})$.

The analysis of Sections 3-4 carries over to parametricity graphs. If $\mathbf{G}$ and $\mathbf{H}$ are parametricity graphs, their exponent $\mathbf{H}^{\mathbf{G}}$ is the exponent reflexive graph with the associated cleavage $[\eta, \eta']\rho$ given by $([\eta, \eta']\rho)(R\colon X \leftrightarrow X') = [\eta_X, \eta'_{X'}](\rho_R)$. The diagonal functor $\Delta : \mathbf{H} \to \mathbf{H}^{\mathbf{G}}$ is cloven. One should verify that in the example reflexive graphs, the parametric limit and colimit functors are cloven.

### 5.5 Representation results

We show that the axiomatization of parametricity graphs supports representation results concerning initial algebras and final coalgebras. A simple case is the following:

**Theorem 13** If $\mathbf{G}$ is a well-pointed, cartesian closed parametricity graph, the parametric limit $\forall_X X \Rightarrow X$ of the functor $\mathrm{Id} \Rightarrow \mathrm{Id}$ is the terminal object.

Note that the theorem only applies to *well-pointed* parametricity graphs. We can show that it breaks in non-well-pointed parametricity graphs. Consider the "presheaf" reflexive graph $\mathbf{REL}^{\mathbf{C}}$ where $\mathbf{C}$ is a category (regarded as a reflexive graph) with a single object $\star$ and a single non-identity arrow $j$ that is idempotent. The terminal object in $\mathbf{REL}^{\mathbf{C}}$ is $\Delta(1)$. If $F\colon \mathbf{C} \to \mathbf{REL}$ is a functor, $(F \Rightarrow F)(\star)$ is a pair $\langle f_{\mathrm{id}}, f_j\rangle$ of functions, each of type $F(\star) \to F(\star)$ subject to naturality conditions. We can find two such pairs, which appear to be given uniformly in $F$:

$$\langle F(\mathrm{id}), F(\mathrm{id})\rangle \quad \text{and} \quad \langle F(j), F(j)\rangle$$

Indeed, there is a parametric transformation $\beta_F\colon \Delta(2) \to (F \Rightarrow F)$ which does not uniquely factor through $\omega_F\colon \Delta(1) \to (F \Rightarrow F)$.

One might draw one of two potential conclusions from this breakdown. One is that parametricity is dependent on well-pointedness of models. (Perhaps, for non-well-pointed models, parametricty does not capture the intuitive idea of uniformity.) A second possible conclusion is that the representation result $\forall_X X \Rightarrow X \cong 1$ is dependent on well-pointedness of models. We tend to the latter view since we have not actually found any examples where parametricity allows intuitively non-uniform expressions. As remarked above, the two distinct elements in $\forall_F F \Rightarrow F$ in $\mathbf{REL}^{\mathbf{C}}$ are both intuitively uniform. Rather, it seems that we do not know at present how to state representation results for non-well-pointed models.

We now turn to the issue of giving general representation results for well-pointed models. Let $F(X)$ denote a type expression of System F with no occurrences of $X$ to the left of '$\to$'. Then the type expression $F(X)$ can be given an interpretation of type $|\mathbf{G}|^n \times \mathbf{G} \to \mathbf{G}$ or, equivalently, $|\mathbf{G}|^n \to \mathbf{G}^{\mathbf{G}}$. Define notations:

$$\begin{array}{rcl} \mu F & \equiv & \forall X.\, (F(X) \to X) \to X \\ \nu F & \equiv & \exists X.\, (X \to F(X)) \times X \end{array}$$

where $\exists X.\, P(X)$ is short-hand for $\forall Y.\, (P(X) \to Y) \to Y$.

**Theorem 14** In any well-pointed parametricity graph model of System F,

- $\mu F$ denotes the initial $F$-algebra, and
- $\nu F$ denotes the final $F$-coalgebra.

Our proof of this result, covered in detail in [4, Chapter 5], is based on giving a formal logic for reasoning about System F terms amounting to an internal logic of well-pointed parametricity graph models. This logic is stronger than System R of Abadi et al. [1] but weaker than the Plotkin-Abadi logic [19]. It has judgments for statements of the form $M\ [R]\ N$ where $M$ and $N$ are terms of System F and $R$ is a relation expression. Relation expressions are built from the syntax:

$$R ::= \alpha \mid \sigma_1 \to \sigma_2 \mid \forall \alpha.\, \sigma \mid {}^{\sigma_0}_{\sigma_1}[M_0, M_1]R$$

where the first three forms, resembling type expressions, stand for the action of type expressions on edges,

and the last form represents the weakest pre-edge construction. Any pure type expression used as a relation, denotes the identity edge of that type. Sample rules in the system are (in a stripped-down notation)

$$\frac{M\ [R \to S]\ N \quad M'\ [R]\ N'}{MM'\ [S]\ NN'}$$

$$\frac{M\ [R]\ N}{\Lambda\alpha.\, M\ [\forall\alpha.\, R]\ \Lambda\alpha.\, N} \qquad \frac{M\ [\forall\alpha.\, R]\ N \quad S\colon \sigma_0 \leftrightarrow \sigma_1}{M[\sigma_0]\ [R[S/\alpha]]\ N[\sigma_1]}$$

$$\frac{MM'\ [R]\ NN'}{M'\ [^{\sigma_0}_{\sigma_1}[M,N]R]\ N'} \qquad \frac{M'\ [^{\sigma_0}_{\sigma_1}[M,N]R]\ N'}{MM'\ [R]\ NN'}$$

Using the rules of the logic, we can prove formal isomorphisms involving System F terms which imply the facts about initial algebras and final coalgebras.

## 6 Conclusion

We have given an abstract characterization of what is meant by a "parametric model" of a polymorphic lambda calculus using the concept of parametric limits in reflexive graph categories. We have also shown that this characterization supports the representation results one expects from parametric models such as the initial algebra and final coalgebra results.

In comparison with the characterization by Ma and Reynolds [13], we might say that the structure required for parametricity is built into the models from the ground up, rather than being imposed after the fact. This leads to a pleasing construction which parallels the presentation of categorical models of simply-typed lambda calculus and other type theories.

Our theory is presented in a generic form addressing parametricity in all its incarnations. In particular, we have shown its applicability to the predicative polymorphic calculus which represents the situation in programming languages like ML and Haskell. The predicative case is interesting because it is representative of the traditional practice in mathemataics, has many classical models, and has useful applications such as the semantics of imperative programming languages [17] or algebraic data types [30].

We have not addressed the issues posed by recursion. Divergence is incompatible with the formal properties of parametricity. However, as observed by Plotkin [21], it is possible for the two to coexist in a polymorphic *linear* lambda calculus. It is possible to tailor the results of this paper to such a calculus, but the details must await a future paper.

## References


[1] M. Abadi, L. Cardelli, and P.-L. Curien. Formal parametric polymorphism. *Theoretical Comput. Sci.*, 121(1-2):9–58, Dec 1993.

[2] E. S. Bainbridge, P. Freyd, A. Scedrov, and P. J. Scott. Functorial polymorphism. *Theoretical Comput. Sci.*, 70:35–64, 1990.

[3] E. W. Dijkstra. *A Discipline of Programming.* Prentice-Hall, Englewood Cliffs, 1976.

[4] B. P. Dunphy. *Parametricity as a Notion of Uniformity in Reflexive Graphs.* PhD thesis, University of Illinois, Dep. of Mathematics, 2002.

[5] M. P. Fiore, A. Jung, E. Moggi, P. W. O'Hearn, J. Riecke, G. Rosolini, and I. Stark. Domains and denotational semantics: History, accomplishments and open problems. *Bulletin of the European Assoc. for Theor. Computer Science*, 59:227–256, 1996.

[6] J.-Y. Girard. *Interpretation Fonctionelle et Elimination des Coupures Dans L'arithemetique D'order Superieur.* PhD thesis, University of Paris, 1972.

[7] David Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Volume II; Extensions of Classical Logic*, pages 497–604. D. Reidel, 1984.

[8] R. Hasegawa. Parametricity of extensionally collapsed term models of polymorphism and their categorical properties. In *Proc. Theoretical Aspects of Computer Software*, volume 526 of *LNCS*. Springer-Verlag, 1991.

[9] C. Hermida. *Fibrations, Logical Predicates and Indeterminantes.* PhD thesis, Aarhus University, 1993.

[10] M. Hyland. The effective topos. In Troelstra and van Dalen, editors, *The Brouwer Symposium.* North-Holland, 1982.

[11] Y. Kinoshita, P. W. O'Hearn, A. J. Power, M. Takeyama, and R. D. Tennent. An axiomatic approach to binary logical relations with applications to data refinement. Manuscript, Queen Mary and Westfield, London, 1997.

[12] G. Longo and E. Moggi. Constructive natural deduction and its 'omega-set' interpretation. *Mathematical Structures in Computer Science*, 1(2):215–254, 1991.

[13] QingMing Ma and J. C. Reynolds. Types, abstraction, and parametric polymorphism, part 2. In *Mathematical Foundations of Programming Semantics, 7th International Conference*, volume 598 of *LNCS*, pages 1–40. Springer-Verlag, 1992.

[14] J. C. Mitchell. *Foundations of Programming Languages*. MIT Press, 1997.

[15] J. C. Mitchell and A. Scedrov. Notes on sconing and relators. In *Computer Science Logic '92, Selected Papers*, volume 702 of *LNCS*, pages 352–378. Springer-Verlag, 1993.

[16] P. W. O'Hearn and J. C. Reynolds. From Algol to polymorphic linear lambda-calculus. *J. ACM*, 47(1):167–223, Jan 2000.

[17] P. W. O'Hearn and R. D. Tennent. Parametricity and local variables. *J. ACM*, 42(3):658–709, 1995. (Reprinted as Chapter 16 of [18]).

[18] P. W. O'Hearn and R. D. Tennent. *Algol-like Languages (Two volumes)*. Birkhäuser, Boston, 1997.

[19] G. Plotkin and M. Abadi. A logic for parametric polymorphism. In *Typed Lambda Calculi and Applications - TLCA '93*, LNCS, pages 361–375. Springer-Verlag, 1993.

[20] G. D. Plotkin. Dijkstra's predicate transformers and Smyth's power domains. In D. Bjorner, editor, *Abstract Software Specifications*, volume 86 of *LNCS*, pages 527–553. Springer-Verlag, 1980.

[21] G. D. Plotkin. Type theory and recursion. In *Proceedings, Eighth Annual IEEE Symposium on Logic in Computer Science*, page 374. IEEE Computer Society Press, June 1993.

[22] U. S. Reddy. When parametricity implies naturality. Electronic manuscript, University of Birmingham, http://www.cs.bham.ac.uk/~udr/, July 1997.

[23] U. S. Reddy. Objects and classes in Algol-like languages. *Information and Computation*, 172:63–97, 2002.

[24] U. S. Reddy and H. Yang. Correctness of data represenations involving heap data structures. In *Programming Languages and Systems: 12th European Symposium on Programming*, volume 2618 of *LNCS*, pages 223–237. Springer-Verlag, 2003.

[25] J. C. Reynolds. Towards a theory of type structure. In *Coll. sur la Programmation*, volume 19 of *LNCS*, pages 408–425. Springer-Verlag, 1974.

[26] J. C. Reynolds. Types, abstraction and parametric polymorphism. In R. E. A. Mason, editor, *Information Processing '83*, pages 513–523. North-Holland, Amsterdam, 1983.

[27] J. C. Reynolds. Polymorphism is not set-theoretic. In G. Kahn, D. B. MacQueen, and G. Plotkin, editors, *Semantics of Data Types*, volume 173 of *LNCS*, pages 145–156. Springer-Verlag, 1984.

[28] John C. Reynolds and Gordon D. Plotkin. On functors expressible in the polymorphic typed lambda calculus. *Information and Computation*, 105(1):1–29, July 1993.

[29] E. Robinson and G. Rosolini. Reflexive graphs and parametric polymorphism. In *Proceedings, Ninth Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, July 1994.

[30] D. Sannella and A. Tarlecki. On observational equivalence and algebraic specification. *J. Comput. Syst. Sci.*, 34:150–178, 1987.

[31] R. A. G. Seely. Categorical semantics for higher order polymorphic lambda calculus. *J. Symbolic Logic*, 52(4):Dec, 969-989 1987.

[32] P. Wadler. Theorems for free! In *Fourth Intern. Conf. Functional Program. Lang. and Computer Architecture*, pages 347–359. ACM, 1989.