INFORMATIQUE THÉORIQUE ET APPLICATIONS

ROLAND BACKHOUSE
PAUL HOOGENDIJK

Final dialgebras : from categories to allegories

Informatique théorique et applications, tome 33, nº 4-5 (1999), p. 401-426
<http://www.numdam.org/item?id=ITA_1999__33_4-5_401_0>

Theoretical Informatics and Applications
Theoret. Informatics Appl. **33** (1999) 401–426

FINAL DIALGEBRAS: FROM CATEGORIES TO ALLEGORIES

ROLAND BACKHOUSE[1] AND PAUL HOOGENDIJK[2]

Abstract. The study of inductive and coinductive types (like finite lists and streams, respectively) is usually conducted within the framework of category theory, which to all intents and purposes is a theory of sets and functions between sets. Allegory theory, an extension of category theory due to Freyd, is better suited to modelling relations between sets as opposed to functions between sets. The question thus arises of how to extend the standard categorical results on the existence of final objects in categories (for example, coalgebras and products) to their existence in allegories. The motivation is to streamline current work on generic programming, in which the use of a relational theory rather than a functional theory has proved to be desirable. In this paper, we define the notion of a relational final dialgebra and prove, for an important class of dialgebras, that a relational final dialgebra exists in an allegory if and only if a final dialgebra exists in the underlying category of maps. Instances subsumed by the class we consider include coalgebras and products. An important lemma expresses bisimulations in allegorical terms and proves this equivalent to Aczel and Mendler's categorical definition.

AMS Subject Classification. 68N05.

1. GENERIC PROGRAMMING

"Generic" or "polytypic" programs are programs that are parameterised by type constructors (functions from types to types, like list) rather than types (like integer or list of integer). Currently there is much effort going into developing

Keywords and phrases: Programming theory, theory of datatypes, inductive type, co-inductive type, relation algebra, category theory, allegory theory, generic programming, polytypic programming.

[1] School of Computer Science and IT, University of Nottingham, Nottingham NG8 1BB, U.K.
[2] Philips Research Laboratories, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands.

both practical applications of generic programming and the underlying theory [5, 7, 10, 14–18, 22]. A major drawback of the current state of the art, however, is that generic programs are typically defined by induction on the structure of the type constructors. This leads to long involved case analyses (for the cases disjoint sum, cartesian product, inductive types etc.). An example can be found in our own work on commuting datatypes [10, 11] in which we give a short semantic account of what it is for two datatypes to commute and then a long case analysis showing that all datatypes in a certain class do indeed commute according to the definition.

The potential benefits of generic programming could be substantially greater if we could lift the level of abstraction one level higher and view all the so-called "regular datatypes" as instances of just one construction. A framework for doing so is evident in Lambek's work on "subequalizers" [20] which later got the name "dialgebra" [9], the name we shall use in this paper. Here we begin an initial exploration of dialgebras as the basic building block of generic programs.

We are concerned with the development of a *relational* as opposed to *functional* theory of generic programming. The reason for this is straightforward: we are interested in specifications as well as implementations. In programming language theory functions on sets play a central role, particularly in so-called "set-theoretic" semantics. There is, for example, a number of well-known, standard set-theoretic results about the existence of initial algebras, and Aczel's work on the anti-foundation axiom [2, 3] is basically about giving a set-theoretic semantics to final coalgebras. The main theorem of this note is about the existence of relational extensions to "set theoretic" (*i.e.* functional) fixpoints.

More formally, the focus of the paper is the existence of final dialgebras in an allegory given their existence in the underlying category of maps. We assume familiarity with elementary category theory. For those unfamiliar with allegory theory, we summarise those elements of the theory that we need for the current discussion. In the words of Freyd and Scedrov [8], allegories are to binary relations between sets as categories are to functions between sets. Thus the focus of the paper is the existence of *relational* extensions to *functional* final dialgebras under the assumption that the functional final dialgebras exist.

We confine our discussion to a particular class of dialgebras, the relevance of which is demonstrated by showing that it includes the class of algebras and coalgebras (thus inductive and coinductive types), as well as sum, product and the unit type (thus non-(co)inductive types). Note that we assume that the reader is familiar with the notions of (initial) algebra and (final) coalgebra. An excellent tutorial introduction to these notions is contained in [13]. We also assume familiarity with fixed point calculus.

Our work draws on two important insights. The first is the class of dialgebras to which we confine the discussion. This insight is inspired by Lambek's discussion of subequalizers [20]. The second is the hylomorphism theorem (see *e.g.* [6] Th. 6.2) which is fundamental to the construction of programs using so-called "virtual" data structures [7, 24].

All the results presented in the paper are easily dualised to initiality rather than finality properties. We have chosen to focus on finality properties for two reasons.

First, some of the concepts we introduce are better known in the context of final algebras – in particular the notion of a bisimulation. (Against this is the fact that we are not aware of references to or applications of the hylomorphism theorem in the context of final coalgebras.) Second, some of the results we obtain are potentially harder to prove in the context of final algebras since dualising results in a relational setting is not always straightforward, unlike in a functional setting, because of the different nature of intersection and union. In the case of the results presented here the process is straightforward.

2. Formal basis

In this section we introduce some basic definitions together with some notation. It is assumed that the reader is familiar with elementary category theory and with fixed point calculus. The notation νf is used to denote the greatest postfix point of monotonic function f. The notation $f : A \leftarrow B$ is used for an arrow f with target object A and source object B in some anonymous category. If we want to be specific about the category, $\mathcal{C}$ say, we write $f : A \overset{\mathcal{C}}{\leftarrow} B$. The application of a functor is denoted by (pre)juxtaposition. Thus, if F is a functor to category $\mathcal{C}$ from category $\mathcal{D}$ and $f : A \overset{\mathcal{D}}{\leftarrow} B$, then $Ff : FA \overset{\mathcal{C}}{\leftarrow} FB$.

An *allegory* is a category with additional structure, the additional structure capturing the most essential characteristics of relations. Being a category means, of course, that for every object A there is an identity arrow id_A, and that every pair of arrows $R : A \leftarrow B$ and $S : B \leftarrow C$, with matching source and target, can be composed: $R \cdot S : A \leftarrow C$. Composition is associative and has id as a unit. The additional axioms include: first, arrows R and S of the same type are ordered by the *partial order* $\subseteq$. Second, their *intersection* (*meet*) $R \cap S$ exists, where $R \cap S$ is defined by the universal property: for all X

$$X \subseteq R \cap S \ \equiv\ X \subseteq R \wedge X \subseteq S.$$

Third, composition is monotonic with respect to the ordering. And fourth, for each arrow $R : A \leftarrow B$ its *converse* $R^\circ : B \leftarrow A$ exists. The converse is required to be its own Galois adjoint: for all Rand S,

$$R \subseteq S^\circ \equiv R^\circ \subseteq S$$

and to commute contravariantly with composition:

$$(R \cdot S)^\circ = S^\circ \cdot R^\circ.$$

It is easily shown from these two laws that converse preserves identities. (When we state laws, such as these, we assume that all variables are appropriately typed.)

The word "relation" in this paper means formally an arrow of an allegory. Occasionally we interpret the laws for binary relations (that is, sets of pairs) in which case the adjective "binary" indicates the particular interpretation.

A relation $R : A \leftarrow B$ is said to be *simple* if $R \cdot R^\circ \subseteq id_A$ and *total* if $R^\circ \cdot R \supseteq id_B$. A relation that is both simple and total is said to be a *map*. (A binary relation $R : A \leftarrow B$ is a map if it is a total function with range A and domain B.) It is easily checked that maps are closed under composition and that identity arrows are maps. Thus the maps of an allegory form a category, which we refer to below as the *underlying map category* of the allegory. Henceforth we denote maps by lower case letters f, g etc. An easily derived rule that we often use is the *shunting rule*: for all maps f, and all relations R and S,

$$f \cdot R \subseteq S \;\equiv\; R \subseteq f^\circ \cdot S.$$

Maps of the same type also have the property that they are equal iff they are comparable. That is, $f \subseteq g \equiv f = g$.

An allegory is said to be *tabulated* if, for each relation $R : A \leftarrow B$, there is a pair of maps $f : A \leftarrow C$ and $g : B \leftarrow C$ such that,

$$R = f \cdot g^\circ \quad \wedge \quad f^\circ \cdot f \cap g^\circ \cdot g \;=\; id_C.$$

(Binary relations are tabulated: take the set C to be R and f and g to be the functions that project a pair onto its left and right components, respectively.)

In an allegory, an object 1 is said to be a *unit* if id_1 is the largest relation of its type and for every object A there exists a total relation $!_A : 1 \leftarrow A$. Together with the first requirement, it follows that $!_A$ is a map. (Simplicity – the requirement that $!_A \cdot !_A^\circ \subseteq id_1$ – follows from $!_A \cdot !_A^\circ : 1 \leftarrow 1$.) If an allegory has a unit then, for all $R : A \leftarrow B$, $R \subseteq !_A^\circ \cdot !_B$. So, for all objects, $!_A^\circ \cdot !_B$ is the greatest relation of type $A \leftarrow B$. An allegory with a unit is said to be *unitary*.

(In a category, an object 1 is a unit object if for each object A there is a unique arrow $!_A : 1 \leftarrow A$. An instance of the general theorems we are about to prove is that an object 1 is a unit in the allegorical sense if and only if it is a unit in the underlying map category. So the notion of a unit in an allegory is the natural extension of the notion of a unit in a category.)

A *partial identity* is a relation X, of type $A \leftarrow A$ for some A, such that $X \subseteq id_A$. In the allegory of binary relations, partial identities represent sets. This is because a pair (x, y) is an element of the partial identity X iff $x = y$; so partial identity $X : A \leftarrow A$ represents the subset of A consisting of those x such that (x, x) is an element of X.

Associated with every relation $R : A \leftarrow B$ in a unitary allegory there are two partial identities, the *right domain* $R{>}$ of R, with $R{>} : B \leftarrow B$ and the *left domain* $R{<}$ of R, with $R{<} : A \leftarrow A$. The right domain operator is defined by the following universal property [1]. For all $X \subseteq id_B$,

$$R \subseteq !_A^\circ \cdot !_B \cdot X \;\equiv\; R{>} \subseteq X. \tag{1}$$

The left domain operator is defined dually. For binary relations, the right domain represents the set of y such that there is at least one x for which $(x,y) \in R$. Proofs of properties of the domain operators that we exploit can be found in [1, 10].

Finally, a *relator* [4] is a monotonic functor that commutes with converse. The identity relator will be denoted by Id. (Bird and De Moor [6] prove that a functor whose domain is a tabulated allegory is monotonic if and only if it commutes with converse. So, in the context of a tabulated allegory, a relator is a monotonic functor. Nevertheless, the property of commuting with converse is so important and, typically, so easily established without recourse to tabularity properties that we prefer to stick with our original definition.)

3. Final dialgebras in a category

(Initial) algebras and (final) coalgebras are well-known concepts. The notion of a dialgebra is a slight generalisation of both notions. In this section we give the formal definition of a final dialgebra in a category and then show that final coalgebras, the unit type and product are all instances of the concept. The section following this one defines the concept in an allegory. We begin with the definition.

3.1. (Categorical) final dialgebras

Suppose F and G are functors of type $\mathcal{B} \leftarrow \mathcal{C}$ for some (possibly different) categories $\mathcal{B}$ and $\mathcal{C}$. Arrow f in $\mathcal{B}$ is an (F,G)-*dialgebra* if $\mathsf{f} : \mathsf{FA} \leftarrow \mathsf{GA}$ for some fixed[1] object A. Object A we call the *carrier* of dialgebra f. Now, α is an (F,G)-(di)homomorphism of type $\mathsf{f} \leftarrow \mathsf{g}$, for (F,G)-dialgebras f and g, iff

$$\mathsf{f} \cdot \mathsf{G}\alpha = \mathsf{F}\alpha \cdot \mathsf{g}.$$

Note that α has type $\mathsf{A} \leftarrow \mathsf{B}$ where A and B are the carriers of dialgebras f and g, respectively.

Defining composition of homomorphisms as the composition in the base category $\mathcal{C}$ and the identity homomorphism on dialgebra f with carrier A as $\mathsf{id_A}$, it is trivial to verify that this defines a category which we denote by $(\mathsf{F},\mathsf{G})\mathsf{DiAlg}$.

An F-algebra is clearly an (Id,F)-dialgebra, and an F-coalgebra is an (F,Id)-dialgebra, where Id denotes the identity functor (on some anonymous category). As is well-known, (co)inductive types can be identified with initial algebras and final coalgebras. The reason we want to generalise to dialgebras is that the non-inductive types are also instances of initial/final dialgebras.

The definition of a final dialgebra we use is standard – a final object in the category of dialgebras – but we give it nonetheless in order to introduce some notation.

[1]Formally, one has to define a dialgebra as the pair (f,A) since F and G need not to be injective.

Definition 2. The arrow $\mathsf{out} : \mathsf{FT} \leftarrow \mathsf{GT}$ is a *final* (F,G)-dialgebra if for each $\mathsf{f} : \mathsf{FA} \leftarrow \mathsf{GA}$ there exists an arrow $[\![\mathsf{f}]\!] : \mathsf{T} \leftarrow \mathsf{A}$ such that for all $\mathsf{h} : \mathsf{T} \leftarrow \mathsf{A}$,

$$\mathsf{h} = [\![\mathsf{f}]\!] \quad \equiv \quad \mathsf{h} : \mathsf{out} \xleftarrow{(\mathsf{F},\mathsf{G})\mathsf{DiAlg}} \mathsf{f}.$$

So, $[\![\mathsf{f}]\!]$ is the unique homomorphism to dialgebra out from dialgebra f. We call $[\![\mathsf{f}]\!]$ the *dianamorphism* of f.

Note that $[\![\mathsf{f}]\!]$ is dependent on F and G. In cases where more than one class of dialgebras is involved we shall write $[\![(\mathsf{F},\mathsf{G})\,;\,\mathsf{f}]\!]$ in order to resolve any ambiguity.

3.2. LIMITS AND COALGEBRAS

In this section we want to show that the universal properties of limits (in particular, unit and product) and coalgebras can all be expressed as a finality property in DiAlg. Expanding the definition of an arrow in DiAlg, Definition 2 becomes

$$\mathsf{h} = [\![\mathsf{f}]\!] \quad \equiv \quad \mathsf{out} \cdot \mathsf{Gh} = \mathsf{Fh} \cdot \mathsf{f}. \tag{3}$$

Our goal is to rewrite the universal properties of limits (in particular unit and product) and final coalgebras as an instance of (3) for a specific choice for out, F and G.

The easier case is the case of final coalgebras – the so-called coinductive datatypes like stream [13]. The universal property of a final coalgebra $\mathsf{out} : \mathsf{FT} \leftarrow \mathsf{T}$ is that, for each $\mathsf{f} : \mathsf{FA} \leftarrow \mathsf{A}$ there exists an arrow $[\![\mathsf{f}]\!] : \mathsf{T} \leftarrow \mathsf{A}$ such that for all $\mathsf{h} : \mathsf{T} \leftarrow \mathsf{A}$,

$$\mathsf{h} = [\![\mathsf{f}]\!] \quad \equiv \quad \mathsf{out} \cdot \mathsf{h} = \mathsf{Fh} \cdot \mathsf{f}. \tag{4}$$

The arrow $[\![\mathsf{f}]\!]$ is called an *anamorphism* [23].

This is clearly an instance of (3). Specifically, equation (4) expresses that out is a final (F,Id)-dialgebra. That is to say, we can rewrite equation (4) as

$$\mathsf{h} = [\![\mathsf{f}]\!] \quad \equiv \quad \mathsf{out} \cdot \mathsf{Id}\,\mathsf{h} = \mathsf{Fh} \cdot \mathsf{f}. \tag{5}$$

or, equivalently,

$$\mathsf{h} = [\![\mathsf{f}]\!] \quad \equiv \quad \mathsf{h} : \mathsf{out} \xleftarrow{(\mathsf{F}\,,\,\mathsf{Id})\mathsf{DiAlg}} \mathsf{f}.$$

Next we consider limits. Let us recall the definition given by Mac Lane [21]. Given categories $\mathcal{C}$ and $\mathcal{D}$, the *diagonal functor* $\Delta : \mathcal{C}^{\mathcal{D}} \leftarrow \mathcal{C}$ sends each object A to the constant functor $\Delta\mathsf{A}$ – the functor which has the value A at each object in $\mathcal{D}$ and the value $\mathsf{id_A}$ at each arrow of $\mathcal{D}$. It sends each arrow f to the constant natural transformation $\Delta\mathsf{f}$ – the natural transformation which has the value f at each object in $\mathcal{D}$. A *limit* for a functor $\mathsf{J} : \mathcal{C} \leftarrow \mathcal{D}$ consists of an object T of $\mathcal{C}$ and a natural transformation $\mathsf{out} : \mathsf{J} \stackrel{\cdot}{\leftarrow} \Delta\mathsf{T}$ which is universal among natural transformations $f : \mathsf{J} \stackrel{\cdot}{\leftarrow} \Delta\mathsf{A}$ for objects A of $\mathcal{C}$. The universal property of out is this: for any natural transformation $\alpha : \mathsf{J} \stackrel{\cdot}{\leftarrow} \Delta\mathsf{A}$ there is a unique arrow $t : \mathsf{T} \leftarrow \mathsf{A}$ such that $\alpha_i = \mathsf{out}_i \cdot t$ for all objects i in $\mathcal{D}$.

It is not too difficult to see that the limit in category $\mathcal{C}$ of the functor $\mathsf{J} : \mathcal{C} \leftarrow \mathcal{D}$ is a final $(\mathsf{K_J}, \Delta)$-dialgebra, where $\mathsf{K_J} : \mathcal{C}^{\mathcal{D}} \leftarrow \mathcal{C}$ is the functor which has the value the functor J at each object A of $\mathcal{C}$ and the value the natural transformation $\mathsf{id_J}$ at each arrow of $\mathcal{C}$. The unique arrow called t by Mac Lane is the arrow $[\![\alpha]\!]$ in our notation. The type of out is $\mathsf{J} \stackrel{\cdot}{\leftarrow} \Delta\mathsf{T}$, which is the same as $\mathsf{K_J T} \leftarrow \Delta\mathsf{T}$. Finally, the universal property of t is the same as the property, for all $h : \mathsf{T} \leftarrow \mathsf{A}$

$$h = [\![\alpha]\!] \quad \equiv \quad \mathsf{out} \cdot \Delta h = \mathsf{K_J} h \cdot \alpha, \tag{6}$$

since $(\Delta h)_i = h$ and $(\mathsf{K_J} h)_i = \mathsf{id}_{\mathsf{J}i}$ for all objects i in $\mathcal{D}$.

It is common to call the category $\mathcal{D}$ the *shape* category. A *unit*[2] is the limit of the empty functor, the unique functor with shape category the empty category $\mathbf{0}$. As is well known, the general definition of a limit given above boils down in the case of a unit to the existence of an object 1 satisfying the requirement that, for all objects A of the category $\mathcal{C}$, there is an arrow $!_{\mathsf{A}} : 1 \leftarrow \mathsf{A}$ such that, for all arrows $h : 1 \leftarrow \mathsf{A}$,

$$h = \,!_{\mathsf{A}}. \tag{7}$$

That is, for all objects A, $!_{\mathsf{A}}$ is the unique object of type $1 \leftarrow \mathsf{A}$. In terms of final dialgebras, out is id_1, the identity arrow on 1, and equation (6) specialises to, for all $h : 1 \leftarrow \mathsf{A}$

$$h = [\![f]\!]. \tag{8}$$

(This is because the implicit universal quantification in the equation

$$\mathsf{out} \cdot \Delta h = \mathsf{K_J} h \cdot \alpha$$

is a quantification over the empty set.) Thus $[\![f]\!]$ is $!_{\mathsf{A}}$ for each $f : \mathsf{A} \leftarrow \mathsf{A}$.

A *product* is the limit of a functor $\mathsf{J} : \mathcal{C} \leftarrow \mathbf{2}$ where $\mathbf{2}$ is the discrete category with two objects. A functor $\mathsf{J} : \mathcal{C} \leftarrow \mathbf{2}$ is a pair of objects (A, B), each of which is in $\mathcal{C}$. The functor $\mathsf{K_J}$ is then the constant functor $\mathsf{K}_{(\mathsf{A},\mathsf{B})}$ that has value the pair (A,B) at each object and the identity function $\mathsf{id}_{(\mathsf{A},\mathsf{B})}$ at each arrow. A natural transformation $\mathsf{out} : \mathsf{J} \stackrel{\cdot}{\leftarrow} \Delta\mathsf{T}$ is a pair of arrows $(\mathsf{outl}, \mathsf{outr})$ where $\mathsf{outl} : \mathsf{A} \leftarrow \mathsf{T}$

[2] The definition given here is the categorical one. We return later to the definition of a unit in an allegory.

and $\mathsf{outr} : \mathsf{B} \leftarrow \mathsf{T}$. Object $\mathsf{A} \times \mathsf{B}$ is the product of A and B if for each $\mathsf{f} : \mathsf{A} \leftarrow \mathsf{C}$ and $\mathsf{G} : \mathsf{B} \leftarrow \mathsf{C}$ there exists an arrow $\mathsf{f} \vartriangle \mathsf{g} : \mathsf{A} \times \mathsf{B} \leftarrow \mathsf{C}$ such that for all $\mathsf{h} : \mathsf{A} \times \mathsf{B} \leftarrow \mathsf{C}$,

$$\mathsf{h} = \mathsf{f} \vartriangle \mathsf{g} \quad \equiv \quad \mathsf{outl} \cdot \mathsf{h} = \mathsf{f} \ \wedge \ \mathsf{outr} \cdot \mathsf{h} = \mathsf{g}. \tag{9}$$

The operator $\vartriangle$ is called the *split* operator. (Sometimes it is also called the fork operator.) In order to match equation (9) with equation (6) we define for $\mathsf{f} : \mathsf{A} \leftarrow \mathsf{C}$ and $\mathsf{g} : \mathsf{B} \leftarrow \mathsf{C}$, $[\![\mathsf{f}\,,\mathsf{g}]\!] \mathrel{\widehat{=}} \mathsf{f} \vartriangle \mathsf{g}$, and state the two equations of the rhs of equation (9) as a single equation in $\mathcal{C}^2$, where $\mathcal{C}$ is the category under consideration. That is, we rewrite equation (9) as

$$\mathsf{h} = [\![\mathsf{f}\,,\mathsf{g}]\!] \ \equiv \ ((\mathsf{outl} \cdot \mathsf{h})\,,(\mathsf{outr} \cdot \mathsf{h})) = (\mathsf{f}\,,\mathsf{g}).$$

Equivalently, using the fact that composition in $\mathcal{C}^2$ is defined componentwise and writing $\Delta\mathsf{h}$ for $(\mathsf{h}\,,\mathsf{h})$,

$$\mathsf{h} = [\![\mathsf{f}\,,\mathsf{g}]\!] \quad \equiv \quad (\mathsf{outl}\,,\mathsf{outr}) \cdot \Delta\mathsf{h} = (\mathsf{f}\,,\mathsf{g}).$$

Collecting all the results, we have shown how the limit of functor J is a final $(\mathsf{K_J}\,,\Delta)$-dialgebra, and a final F-coalgebra is a final $(\mathsf{F}\,,\mathrm{Id})$-dialgebra. Dually the colimit of functor J is an initial $(\Delta\,,\mathsf{K_J})$-dialgebra, and an initial F-algebra is an initial $(\mathrm{Id}\,,\mathsf{F})$-dialgebra.

4. Relational final dialgebras

In this section, we suggest a definition for a relational extension of final dialgebras and prove some of its properties. Thus, whereas in the last section the context of our discussion was category theory, in this section it is allegory theory.

Note that, if $\mathcal{D}$ is a discrete category and $\mathcal{C}$ is an allegory, $\mathcal{C}^{\mathcal{D}}$ is an allegory in which the allegorical operations (composition, converse, subset etc.) are defined componentwise. Also $\Delta : \mathcal{C}^{\mathcal{D}} \leftarrow \mathcal{C}$ is a relator (we leave the simple verification to the reader) and $\mathsf{K_J}$ is a relator for each functor $\mathsf{J} : \mathcal{C} \leftarrow \mathcal{D}$. Thus, discrete limits in a category $\mathcal{C}$ are final (F,G)-dialgebras where both F and G are relators, provided that $\mathcal{C}$ is an allegory. Also, if $\mathcal{C}$ is an allegory then obviously the identity functor on $\mathcal{C}$ is a relator. So if endofunctor F on $\mathcal{C}$ is a relator a final F-coalgebra is a final (F,G)-dialgebra where, again, both F and G are relators.

Note, also, that *local completeness* of allegory $\mathcal{C}$ is the requirement that, for each pair of objects A and B, the partially ordered set of arrows of type $\mathsf{A} \leftarrow \mathsf{B}$ is complete. This, by definition of completeness, is the requirement that, for all discrete categories $\mathcal{D}$, the relator $\Delta : \mathcal{C}^{\mathcal{D}} \leftarrow \mathcal{C}$ is a *lower adjoint* in a Galois connection. The relator G is a *lower adjoint* in a Galois connection if, for all objects A and B and relation $\mathsf{R} : \mathsf{GA} \leftarrow \mathsf{GB}$, there is a relation $\mathsf{G}^\sharp\mathsf{R} : \mathsf{A} \leftarrow \mathsf{B}$ such that, for all relations $\mathsf{X} : \mathsf{A} \leftarrow \mathsf{B}$,

$$\mathsf{GX} \subseteq \mathsf{R} \equiv \mathsf{X} \subseteq \mathsf{G}^\sharp\mathsf{R}.$$

In the case of $G = \Delta : \mathcal{C}^{\mathcal{D}} \leftarrow \mathcal{C}$ the upper adjoint is the infimum operator of shape $\mathcal{D}$. For example, if $\mathcal{D} = \mathbf{2}$ then the upper adjoint is binary intersection. That is,

$$\Delta X \subseteq (R, S) \equiv X \subseteq R \cap S$$

where the ordering between ΔX and (R, S) is componentwise. In the general case, for each relation $S : \Delta A \leftarrow \Delta B$ (*i.e.* family of relations $S_i : A \leftarrow B$ indexed by objects i of $\mathcal{D}$) the infimum of S is the relation $\cap S$ satisfying, for all $X : A \leftarrow B$,

$$\Delta X \subseteq S \equiv X \subseteq \cap S,$$

i.e.

$$\forall \langle i : i \in \mathcal{D} : X \subseteq S_i \rangle \equiv X \subseteq \cap S.$$

These two observations are the basis for our being able to discuss limits and coalgebras simultaneously. From now on we consider final (F,G)-dialgebras where both F and G are relators, and G is a lower adjoint in a Galois connection, with upper adjoint $G^\sharp$. (Note: we do not assume that $G^\sharp$ is a relator.)

In fact, the assumption that G is a lower adjoint is not strictly necessary until Section 6, which is the first place that we assume the local completeness of the allegory. Some of the results in earlier sections can, however, be made sharper if we assume that the relator G is a lower adjoint. Thus, up until Section 6 we make the assumption explicit in any results where it is used; in Section 6, on the other hand, it is a global assumption.

The focus of the paper is the following definition which we propose as the natural extension of the categorical notion of a final dialgebra to an allegorical notion.

Definition 10. Assume that F and G are relators of the same type. Then (T, out) is a *relational final* (F, G)-*dialgebra* iff $\mathsf{out} : FT \leftarrow GT$ is a simple (F, G)-dialgebra and there is a mapping $[\![\_]\!]$ defined on all (F, G)-dialgebras such that

$$[\![R]\!] : T \leftarrow A \text{ if } R : FA \leftarrow GA, \tag{11}$$

$$[\![\mathsf{out}]\!] = \mathsf{id}_T \text{ , and} \tag{12}$$

$$[\![R]\!]^\circ \cdot [\![S]\!] \text{ is the largest solution of the equation } X{::}\ GX \subseteq R^\circ \cdot FX \cdot S. \tag{13}$$

The mapping $[\![\_]\!]$ we call the *relational dianamorphism*. Properties (12) and (13), we call the *reflection law* and *cancellation law*, respectively. □

The key insight in this definition is property (13) which is a slight generalisation of the dual of the hylomorphism theorem for initial algebras (see *e.g.* [6] Th. 6.2). What we have done here is to elevate it from being a theorem to being a requirement.

Our goal now is to provide evidence that this definition is indeed a natural extension. The remainder of this section is concerned with proving that a relational final dialgebra is the categorical final dialgebra in the underlying map category. Section 6 establishes a converse of this result. We begin with some simple lemmas.

First, it is useful to restate the property (13) as a pair of calculational rules. These are, first $[\![R]\!]^\circ \cdot [\![S]\!]$ is a solution of the given equation: for all R and S,

$$\mathsf{G}([\![\mathsf{R}]\!]^\circ \cdot [\![\mathsf{S}]\!]) \subseteq \mathsf{R}^\circ \cdot \mathsf{F}([\![\mathsf{R}]\!]^\circ \cdot [\![\mathsf{S}]\!]) \cdot \mathsf{S}, \tag{14}$$

and, second, it is at least any other solution: for all X, R and S,

$$\mathsf{X} \subseteq [\![\mathsf{R}]\!]^\circ \cdot [\![\mathsf{S}]\!] \;\Leftarrow\; \mathsf{GX} \subseteq \mathsf{R}^\circ \cdot \mathsf{FX} \cdot \mathsf{S}. \tag{15}$$

Second, it is useful to combine the cancellation law (13) with the reflection law (12). Straightforward calculation, with S instantiated to out and using the properties of converse, gives the simplifications:

$$\mathsf{G}[\![\mathsf{R}]\!] \subseteq \mathsf{out}^\circ \cdot \mathsf{F}[\![\mathsf{R}]\!] \cdot \mathsf{R}, \tag{16}$$

and

$$\mathsf{X} \subseteq [\![\mathsf{R}]\!] \;\Leftarrow\; \mathsf{GX} \subseteq \mathsf{out}^\circ \cdot \mathsf{FX} \cdot \mathsf{R}. \tag{17}$$

The properties (14) and (16) we call the *computation* rules, and properties (15) and (17) we call the *coinduction* rules. As a simple application of these rules, we show that the function $[\![\_]\!]$ is monotonic.

$$\begin{array}{ll}
 & [\![\mathsf{R}]\!] \subseteq [\![\mathsf{S}]\!] \\
\Leftarrow & \quad \{ \quad (17) \text{ with } \mathsf{X},\mathsf{R} := [\![\mathsf{R}]\!],\mathsf{S} \ \} \\
 & \mathsf{G}[\![\mathsf{R}]\!] \subseteq \mathsf{out}^\circ \cdot \mathsf{F}[\![\mathsf{R}]\!] \cdot \mathsf{S} \\
\Leftarrow & \quad \{ \quad (16) \text{ and transitivity of } \subseteq \ \} \\
 & \mathsf{out}^\circ \cdot \mathsf{F}[\![\mathsf{R}]\!] \cdot \mathsf{R} \subseteq \mathsf{out}^\circ \cdot \mathsf{F}[\![\mathsf{R}]\!] \cdot \mathsf{S} \\
\Leftarrow & \quad \{ \quad \text{monotonicity of composition } \} \\
 & \mathsf{R} \subseteq \mathsf{S}\ .
\end{array}$$

We remarked above that, in the cases we are particularly interested in, G is a lower adjoint. With this assumption, the inclusion

$$\mathsf{GX} \subseteq \mathsf{R}^\circ \cdot \mathsf{FX} \cdot \mathsf{S}$$

is equivalent to

$$\mathsf{X} \subseteq \mathsf{G}^\sharp(\mathsf{R}^\circ \cdot \mathsf{FX} \cdot \mathsf{S}).$$

So, if G is a lower adjoint with upper adjoint $G^\sharp$, we have:

$$[\![R]\!]^\circ \cdot [\![S]\!] = \nu(X \mapsto G^\sharp(R^\circ \cdot FX \cdot S)) \tag{18}$$

and

$$[\![R]\!] = \nu(X \mapsto G^\sharp(\mathsf{out}^\circ \cdot FX \cdot R)). \tag{19}$$

It is informative to instantiate F and G with the values we obtained for them when discussing the product functor in Section 3.2. Thus, for G we take the doubling relator Δ and for F we take $K_{(A,B)}$. Then $G^\sharp$ is the binary intersection operator. (That is, $G^\sharp(R,S) = R \cap S$.) Also, supposing that $\mathsf{out}_{(A,B)} = (\mathsf{outl}_{(A,B)}, \mathsf{outr}_{(A,B)})$, we obtain that

$$\begin{array}{ll} & [\![R,S]\!] \\ = & \qquad \{ \quad (19),\ \text{definitions of } F,\ G^\sharp \text{ and composition } \} \\ & \mathsf{outl}^\circ \cdot R \cap \mathsf{outr}^\circ \cdot S . \end{array}$$

This is the standard definition of $R \vartriangle S$, the extension to relations of the split operator discussed in Section 3.2.

We now return to the general case.

Lemma 20. $[\![\_]\!]$ preserves simple relations and total relations. Hence $[\![\_]\!]$ preserves maps.

Proof. Let $f : FA \leftarrow GA$ be simple. Then:

$$\begin{array}{ll} & [\![f]\!] \cdot [\![f]\!]^\circ \subseteq \mathsf{id}_T \\ \equiv & \qquad \{ \quad \text{reflection: } (12) \ \} \\ & [\![f]\!] \cdot [\![f]\!]^\circ \subseteq [\![\mathsf{out}]\!] \\ \Leftarrow & \qquad \{ \quad \text{coinduction } (17) \ \} \\ & G([\![f]\!] \cdot [\![f]\!]^\circ) \subseteq \mathsf{out}^\circ \cdot F([\![f]\!] \cdot [\![f]\!]^\circ) \cdot \mathsf{out} \\ \Leftarrow & \qquad \{ \quad G \text{ is a relator, computation } (16);\ F \text{ is a relator } \} \\ & \mathsf{out}^\circ \cdot F[\![f]\!] \cdot f \cdot f^\circ \cdot F[\![f]\!]^\circ \cdot \mathsf{out} \subseteq \mathsf{out}^\circ \cdot F[\![f]\!] \cdot F[\![f]\!]^\circ \cdot \mathsf{out} \\ \Leftarrow & \qquad \{ \quad \text{monotonicity } \} \\ & f \cdot f^\circ \subseteq \mathsf{id}_{FA} \\ \equiv & \qquad \{ \quad f \text{ is simple } \} \\ & \mathsf{true} . \end{array}$$

Let $R : FA \leftarrow GA$ be total. That is, $\mathsf{id}_{GB} \subseteq R^\circ \cdot R$. Then:

$$
\begin{array}{ll}
 & \mathsf{id_A} \subseteq [\![\mathsf{R}]\!]^\circ \cdot [\![\mathsf{R}]\!] \\
\Leftarrow & \quad \{ \quad \text{coinduction (15)} \} \\
 & \mathsf{Gid_A} \subseteq \mathsf{R}^\circ \cdot \mathsf{Fid_A} \cdot \mathsf{R} \\
\equiv & \quad \{ \quad \mathsf{F} \text{ is a relator} \} \\
 & \mathsf{Gid_A} \subseteq \mathsf{R}^\circ \cdot \mathsf{R} \\
\equiv & \quad \{ \quad \mathsf{G} \text{ is a relator, } \mathsf{R} \text{ is total} \} \\
 & \mathsf{true}\ .
\end{array}
$$

□

Next we aim to prove that out is a map. The assumption is that out is simple so we only have to prove that out is total.

Lemma 21. out is a total relation.

Proof.

$$
\begin{array}{ll}
 & \mathsf{out}^\circ \cdot \mathsf{out} \\
= & \quad \{ \quad \text{reflection: (12)} \} \\
 & \mathsf{out}^\circ \cdot \mathsf{F}[\![\mathsf{out}]\!] \cdot \mathsf{out} \\
\supseteq & \quad \{ \quad \text{computation (16)} \} \\
 & \mathsf{G}\,[\![\mathsf{out}]\!] \\
= & \quad \{ \quad \mathsf{id_{GT}} = \mathsf{Gid_T}\text{, reflection: (12)} \} \\
 & \mathsf{id_{GT}}\ .
\end{array}
$$

□

In the case that G is a lower adjoint we can prove a stronger statement than Lemma 21, viz:

Lemma 22. If relator G is a lower adjoint with upper adjoint $\mathsf{G}^\sharp$ then $\mathsf{G}^\sharp(\mathsf{out}^\circ \cdot \mathsf{out}) = \mathsf{id_T}$.

Proof. From Lemma 21 we know, by monotonicity of $\mathsf{G}^\sharp$, that

$$\mathsf{G}^\sharp(\mathsf{out}^\circ \cdot \mathsf{out}) \supseteq \mathsf{G}^\sharp\mathsf{id_{GT}} \supseteq \mathsf{id_T}.$$

So it suffices to prove the opposite inclusion.

$$
\begin{array}{ll}
 & \mathsf{G}^\sharp(\mathsf{out}^\circ \cdot \mathsf{out}) \subseteq \mathsf{id_T} \\
\equiv & \quad \{ \quad \text{reflection: (12)} \} \\
 & \mathsf{G}^\sharp(\mathsf{out}^\circ \cdot \mathsf{out}) \subseteq [\![\mathsf{out}]\!] \\
\Leftarrow & \quad \{ \quad \text{coinduction (17)} \} \\
 & \mathsf{G}^\sharp(\mathsf{out}^\circ \cdot \mathsf{out}) \subseteq \mathsf{G}^\sharp(\mathsf{out}^\circ \cdot \mathsf{FG}^\sharp(\mathsf{out}^\circ \cdot \mathsf{out}) \cdot \mathsf{out})
\end{array}
$$

$$
\begin{array}{ll}
\Leftarrow & \{ \quad \text{monotonicity} \} \\
& \mathsf{id}_{\mathsf{FT}} \subseteq \mathsf{F}\mathsf{G}^{\sharp}(\mathsf{out}^{\circ} \cdot \mathsf{out}) \\
\equiv & \{ \quad \mathsf{F} \text{ is a relator, Lem. 21} \} \\
& \mathsf{true}\ .
\end{array}
$$

□

In the case of product, Lemma 22 is the property that

$$\mathsf{outl}^{\circ} \cdot \mathsf{outl} \cap \mathsf{outr}^{\circ} \cdot \mathsf{outr} \;=\; \mathsf{id}_{\mathsf{A}\times\mathsf{B}}$$

where $(\mathsf{outl}, \mathsf{outr})$ is a final $(\mathsf{K}_{(\mathsf{A},\mathsf{B})}, \Delta)$ dialgebra. In the case of unit this is the trivial property that $\mathsf{id}_1{}^{\circ} \cdot \mathsf{id}_1 = \mathsf{id}_1$. In the case of F-coalgebras, G is the identity relator, as is $\mathsf{G}^{\sharp}$. Thus in this case we get that $\mathsf{out}^{\circ} \cdot \mathsf{out} = \mathsf{id}_{\mathsf{T}}$. This is half way towards proving Lambek's Lemma [19], namely that if G is the identity relator then out is an isomorphism. Here are the details of the remainder of the proof.

Lemma 23. If G is the identity relator then $\mathsf{out}^{\circ} = [\![\mathsf{Fout}]\!]$.

Proof.

$$
\begin{array}{ll}
& \mathsf{out}^{\circ} \\
= & \{ \quad \text{In preparation for using reflection,} \\
& \qquad \text{we introduce the identity } \mathsf{id}_{\mathsf{T}}\text{. Specifically,} \\
& \qquad \text{since F is a functor, } \mathsf{Fid}_{\mathsf{T}} = \mathsf{id}_{\mathsf{FT}} \} \\
& \mathsf{out}^{\circ} \cdot \mathsf{Fid}_{\mathsf{T}} \\
= & \{ \quad \text{reflection: (12)} \} \\
& \mathsf{out}^{\circ} \cdot \mathsf{F}[\![\mathsf{out}]\!] \\
= & \{ \quad \text{definition of dianamorphism} \} \\
& \mathsf{out}^{\circ} \cdot \mathsf{F}\nu(\mathsf{X} \mapsto \mathsf{out}^{\circ} \cdot \mathsf{FX} \cdot \mathsf{out}) \\
= & \{ \quad \text{rolling rule, associativity of composition} \} \\
& \nu(\mathsf{X} \mapsto \mathsf{out}^{\circ} \cdot \mathsf{F}(\mathsf{X} \cdot \mathsf{out})) \\
= & \{ \quad \mathsf{F} \text{ is a relator} \} \\
& \nu(\mathsf{X} \mapsto \mathsf{out}^{\circ} \cdot \mathsf{FX} \cdot \mathsf{Fout}) \\
= & \{ \quad \text{(19), } \mathsf{G}^{\sharp} \text{ is the identity function} \} \\
& [\![\mathsf{Fout}]\!]\ .
\end{array}
$$

□

Lemma 24. If G is the identity relator then $\mathsf{out} \cdot \mathsf{out}^{\circ} = \mathsf{id}_{\mathsf{FT}}$.

Proof. From Lemma 23 it follows that

$$\mathsf{out} \cdot \mathsf{out}^{\circ} = [\![\mathsf{Fout}]\!]^{\circ} \cdot [\![\mathsf{Fout}]\!]\,.$$

We continue with the rhs:

$$
\begin{array}{ll}
 & [\![\mathsf{Fout}]\!]^{\circ} \cdot [\![\mathsf{Fout}]\!] \\
= & \quad \{ \quad \text{cancellation: (18)} \ \} \\
 & \nu(\mathsf{X} \mapsto \mathsf{Fout}^{\circ} \cdot \mathsf{FX} \cdot \mathsf{Fout}) \\
= & \quad \{ \quad \mathsf{F} \text{ is a functor} \ \} \\
 & \nu(\mathsf{X} \mapsto \mathsf{F}(\mathsf{out}^{\circ} \cdot \mathsf{X} \cdot \mathsf{out})) \\
= & \quad \{ \quad \text{rolling rule} \ \} \\
 & \mathsf{F}\nu(\mathsf{X} \mapsto \mathsf{out}^{\circ} \cdot \mathsf{FX} \cdot \mathsf{out}) \\
= & \quad \{ \quad \text{(19), } \mathsf{G}^{\sharp} \text{ is the identity function} \ \} \\
 & \mathsf{F}[\![\mathsf{out}]\!] \\
= & \quad \{ \quad \text{reflection: (12), } \mathsf{F} \text{ is a functor} \ \} \\
 & \mathsf{id}_{\mathsf{FT}} \ .
\end{array}
$$

□

Note that Lemma 24 shows that the requirement that out is simple in Definition 10 is superfluous in the case that G is the identity relator. It is straightforward to prove that out is simple in the case of product [1] but we know of no way of deriving the property from the remaining requirements in the definition.

Corollary 25. If G is the identity relator then out is an isomorphism. Moreover,

$$\mathsf{out} \cdot [\![\mathsf{R}]\!] = \mathsf{F}[\![\mathsf{R}]\!] \cdot \mathsf{R}. \tag{26}$$

Proof. That out is an isomorphism is a combination of Lemmas 24 and 23. Property (26) follows from (19) and the fact that out is an isomorphism. □

We conclude this section with a summarising theorem.

Theorem 27. A relational final dialgebra is the categorical final dialgebra in the underlying map category. That is, for maps h and f, $[\![\mathsf{f}]\!]$ is also a map and satisfies the unique extension property

$$\mathsf{h} = [\![\mathsf{f}]\!] \ \equiv \ \mathsf{out} \cdot \mathsf{Gh} = \mathsf{Fh} \cdot \mathsf{f}.$$

Proof.

$$
\begin{array}{ll}
 & \mathsf{out} \cdot \mathsf{Gh} = \mathsf{Fh} \cdot \mathsf{f} \\
\equiv & \quad \{ \quad \text{all components are maps, shunting} \ \} \\
 & \mathsf{Gh} \subseteq \mathsf{out}^{\circ} \cdot \mathsf{Fh} \cdot \mathsf{f} \\
\Rightarrow & \quad \{ \quad \text{coinduction (17)} \ \} \\
 & \mathsf{h} \subseteq [\![\mathsf{f}]\!]
\end{array}
$$

$$
\begin{array}{ll}
\equiv & \{\quad \text{by Lem. 20 } [\![f]\!] \text{ is a map } \} \\
& h = [\![f]\!] \\
\Rightarrow & \{\quad \text{computation (16) } \} \\
& Gh \subseteq out^{\circ} \cdot Fh \cdot f \\
\equiv & \{\quad \text{all components are maps, shunting } \} \\
& out \cdot Gh = Fh \cdot f\ .
\end{array}
$$

Hence,

$$h = [\![f]\!] \ \equiv\ out \cdot Gh = Fh \cdot f\ .$$

□

5. Parameterized final data structures

The construction of (co)inductive type structures like *List* and *Stream* is now well-understood. The general procedure for constucting a coinductive type is to take a binary relator, $\otimes$ say, fix one of its arguments, to say A (thus consider the relator $(A\otimes)$), construct the final $(A\otimes)$-coalgebra, and finally abstract from A. In this section we generalise this construction method to dialgebras, and show that, with one and the same theorem, we can prove that coinductive types and product are relators.

Theorem 28. Suppose $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are allegories. Suppose $\otimes$ is a binary relator of type $\mathcal{B} \leftarrow \mathcal{A}\times\mathcal{C}$ and G is a relator of type $\mathcal{B} \leftarrow \mathcal{C}$. Suppose also that, for each object A in $\mathcal{A}$, $out_A \ :\ A\otimes TA \leftarrow GTA$ is a relational final $((A\otimes), G)$-dialgebra. Then, for relation $R : A \leftarrow B$ in allegory $\mathcal{A}$,

$$TR \ \hat{=}\ [\![((A\otimes), G)\, ;\ R\otimes id_{TB} \cdot out_B]\!]$$

defines a relator, T, of type $\mathcal{C} \leftarrow \mathcal{A}$.

Proof. The monotonicity of T is obvious since it is defined as the functional composition of a number of functions ($[\![\_]\!]$, $\_\otimes id$ and $\_\cdot out$) that are all monotonic. For commuting with converse, we calculate

$$
\begin{array}{ll}
& (TR)^{\circ} \subseteq T(R^{\circ}) \\
\Leftarrow & \{\quad \text{coinduction (17) } \} \\
& G(TR)^{\circ} \subseteq out^{\circ} \cdot id\otimes(TR)^{\circ} \cdot R^{\circ}\otimes id \cdot out \\
\equiv & \{\quad \text{converse, } G \text{ and } \otimes \text{ relators } \} \\
& GTR \subseteq out^{\circ} \cdot R\otimes TR \cdot out \\
\equiv & \{\quad \text{computation (16) } \} \\
& true\ .
\end{array}
$$

Instantiating $\mathsf{R} := \mathsf{R}^\circ$ and taking converses, gives $\mathsf{T}(\mathsf{R}^\circ) \subseteq (\mathbb{T}\mathsf{R})^\circ$. Hence $(\mathbb{T}\mathsf{R})^\circ = \mathsf{T}(\mathsf{R}^\circ)$. The fact that T respects identities follows directly from the reflection law (12).

Distribution over composition follows from the following so-called *map fusion* rule[3]:

$$\mathbb{T}\mathsf{R} \cdot [\![\mathsf{S}]\!] = [\![\mathsf{R}{\otimes}\mathsf{id} \cdot \mathsf{S}]\!]$$

since

$$
\begin{array}{ll}
 & \mathbb{T}\mathsf{R} \cdot \mathbb{T}\mathsf{S} \\
= & \qquad \{ \quad \text{definition } \mathsf{T} \,\} \\
 & \mathbb{T}\mathsf{R} \cdot [\![\mathsf{S}{\otimes}\mathsf{id} \cdot \mathsf{out}]\!] \\
= & \qquad \{ \quad \text{map fusion rule} \,\} \\
 & [\![\mathsf{R}{\otimes}\mathsf{id} \cdot \mathsf{S}{\otimes}\mathsf{id} \cdot \mathsf{out}]\!] \\
= & \qquad \{ \quad \otimes \text{ functor} \,\} \\
 & [\![(\mathsf{R}\cdot\mathsf{S}){\otimes}\mathsf{id} \cdot \mathsf{out}]\!] \\
= & \qquad \{ \quad \text{definition } \mathsf{T} \,\} \\
 & \mathsf{T}(\mathsf{R}\cdot\mathsf{S}) \ .
\end{array}
$$

The map fusion rule, we prove as follows. First, since $\mathbb{T}\mathsf{R} = (\mathsf{T}(\mathsf{R}^\circ))^\circ$, we have by definition of T:

$$\mathbb{T}\mathsf{R} \cdot [\![\mathsf{S}]\!] \;=\; [\![\mathsf{R}^\circ{\otimes}\mathsf{id}\cdot\mathsf{out}]\!]^\circ \cdot [\![\mathsf{S}]\!] .$$

So $\mathbb{T}\mathsf{R} \cdot [\![\mathsf{S}]\!]$ is the largest solution of the equation

$$\mathsf{X}{::}\ \ \mathsf{GX} \subseteq (\mathsf{R}^\circ{\otimes}\mathsf{id}\cdot\mathsf{out})^\circ \cdot \mathsf{id}{\otimes}\mathsf{X} \cdot \mathsf{S}.$$

Now $[\![\mathsf{R}{\otimes}\mathsf{id} \cdot \mathsf{S}]\!]$ is the largest solution of the equation

$$\mathsf{X}{::}\ \ \mathsf{GX} \subseteq \mathsf{out}^\circ \cdot \mathsf{id}{\otimes}\mathsf{X} \cdot \mathsf{R}{\otimes}\mathsf{id} \cdot \mathsf{S}.$$

[3]The use of "map" here has a different meaning to elsewhere in the paper. It alludes to the "map" function on lists.

But these two equations are identical since, for all X and R,

$$
\begin{array}{cl}
 & (R^\circ \otimes id \cdot out)^\circ \cdot id \otimes X \\
= & \{ \quad \text{distribution of converse over composition } \} \\
 & out^\circ \cdot (R^\circ \otimes id)^\circ \cdot id \otimes X \\
= & \{ \quad \text{converse distributes over binary relator } \otimes, (R^\circ)^\circ = R \ \} \\
 & out^\circ \cdot R \otimes id \cdot id \otimes X \\
= & \{ \quad \otimes \text{ is a binary functor } \} \\
 & out^\circ \cdot id \otimes X \cdot R \otimes id \ .
\end{array}
$$

Thus $\mathbb{T}R \cdot [\![S]\!]$ and $[\![R \otimes id \cdot S]\!]$ are the largest solutions of identical equations and thus are equal. □

We call relators constructed as above *coregular.*

An elementary example of this theorem is obtained by defining $\otimes$ to be the projection relator *Exl* (that is, $X \otimes Y = X$ for both objects and arrows), G to be the identity relator and out_A to be id_A. Then $(A\otimes) = K_A$ and, for $R : A \leftarrow B$ $(= A \otimes B \leftarrow GB)$,

$$
\begin{array}{cl}
 & true \\
\equiv & \{ \quad (25), \text{ G is the identity relator } \} \\
 & out_A \cdot [\![R]\!] = F[\![R]\!] \cdot R \\
\equiv & \{ \quad \text{definitions of F and } out_A \ \} \\
 & [\![R]\!] = R \ .
\end{array}
$$

Thus,

$$
\begin{array}{cl}
 & \mathbb{T}R \\
= & \{ \quad \text{definition of T } \} \\
 & [\![R \otimes id_{TB} \cdot out_B]\!] \\
= & \{ \quad \text{definitions of } \otimes \text{ and } out_B \ \} \\
 & [\![R]\!] \\
= & \{ \quad \text{above } \} \\
 & R \ .
\end{array}
$$

Thus the identity relator is coregular.

The product relator is also coregular. Again we take $\otimes$ to be the projection relator *Exl* but now on a product allegory. Thus $(U,V)\otimes(X,Y) \mathrel{\widehat{=}} (U,V)$. For G we take the doubling relator Δ. Thus $((A,B)\otimes) = K_{(A,B)}$ and $G^\sharp$ is the binary intersection operator. We showed earlier (see the discussion immediately following (19)) that with this choice $[\![R,S]\!] = R \vartriangle S = outl^\circ \cdot R \cap outr^\circ \cdot S$. It thus follows that

$$\mathsf{T}(\mathsf{R},\mathsf{S}) \;=\; (\mathsf{R}\cdot\mathsf{outl})\vartriangle(\mathsf{S}\cdot\mathsf{outl}) \;=\; \mathsf{outl}^\circ\cdot\mathsf{R}\cdot\mathsf{outl}\cap\mathsf{outr}^\circ\cdot\mathsf{S}\cdot\mathsf{outr}$$

which is the standard definition of $\mathsf{R}\times\mathsf{S}$, the extension of the product functor to relations [1,6]. The map fusion theorem is the product-split fusion theorem:

$$\mathsf{R}\times\mathsf{S}\;\cdot\;\mathsf{T}\vartriangle\mathsf{U} \;=\; (\mathsf{R}\cdot\mathsf{T})\vartriangle(\mathsf{S}\cdot\mathsf{U}).$$

Having shown how the product relator is constructed we can conclude with possibly the best known example, namely streams. Taking the relator $\otimes$ to be product, the relator T constructed as above is *Stream*. That is, TA is the type of all infinite sequences of A's and TR is a relation holding between two streams if and only corresponding elements of the two streams are related by R.

6. Existence of relational dianamorphisms

In the previous section we showed that the relational dianamorphism is an extension of the dianamorphism on maps. In other words, if a relational final dialgebra exists, then it is also a final dialgebra in the underlying map category. In this section we show that, under some reasonable conditions on the allegory, the other way around is also true.

The context in which our main theorem holds is as follows. We assume that $\mathcal{C}$ is a locally complete, tabular allegory. We also assume that $\mathcal{D}$ is a discrete category, and F and G are relators of type $\mathcal{C}^{\mathcal{D}}\leftarrow\mathcal{C}$. We refer to $\mathcal{C}$ as "the allegory" and $\mathcal{D}$ as "the shape category". (In the case of coalgebras $\mathcal{D}$ is, of course, $\mathbf{1}$, the category with exactly one object and one arrow.) Finally, we assume the axiom of choice, viz. below each total relation there is a map. To be precise:

$$\mathsf{R}\text{ is total} \;\equiv\; \exists(\mathsf{f} : \text{map }\mathsf{f} : \;\mathsf{f}\subseteq\mathsf{R}). \tag{29}$$

For allegory Rel, the allegory of binary relations between sets, all of these assumptions hold.

The theorem we prove is that if there is a final (F,G)-dialgebra in the sub-category of $\mathcal{C}$ formed by the maps then it is also relational according to Definition 10. For the unit and product this is already known: for a tabular allegory, the relational extensions of unit and product exist precisely when unit and product exist for the sub-category of maps. Our contribution is to show how this is proved for dialgebras in general, with particular instances limits and coalgebras. In summary form, this is the theorem we are about to prove:

Theorem 30. If $\mathsf{out} : \mathsf{FT}\leftarrow\mathsf{GT}$ is a final dialgebra for the sub-category of maps then out is a *relational* final dialgebra.

The remainder of this section is devoted to proving this theorem. In order to show that out is a relational final dialgebra, we have to define the relational extension of $[\![\_]\!]$, the dianamorphism operator on maps. In view of (16) and (17) the only possible candidate is the function mapping relation $R : FA \leftarrow GA$ to the largest solution of the equation

$$X\text{::}\ GX \subseteq out^{\circ} \cdot FX \cdot R.$$

Let us denote this function by $[\_]$. Thus, by definition,

$$G[R] \subseteq out^{\circ} \cdot F[R] \cdot R, \tag{31}$$

and

$$X \subseteq [R] \Leftarrow GX \subseteq out^{\circ} \cdot FX \cdot R. \tag{32}$$

It is in order to guarantee the existence of the function $[\_]$ that we need to assume that the allegory is locally complete, and G is a lower adjoint. If these two assumptions hold then $[R]$ is the greatest postfix point of the function mapping relation $X : T \leftarrow A$ to $G^{\sharp}(out^{\circ} \cdot FX \cdot R)$, which (by the well-known Knaster-Tarski theorem) exists by virtue of the completeness of $T \leftarrow A$.

Let us now turn to the algebraic properties demanded of the function $[\_]$. The typing law (11) is clearly satisfied. We have to show that the reflection and the cancellation laws ((12) and (13)) hold for the mapping $[\_]$. We also have to show that $[\_]$ extends $[\![\_]\!]$, *i.e.* the two functions coincide when applied to maps. We start with the reflection law. Because it proves useful in other contexts we introduce the following definition:

Definition 33 (Bisimulation). Suppose $k : FA \leftarrow GA$ is an (F,G)-dialgebra. A relation R of type $A \leftarrow A$ is a *bisimulation* of k if

$$GR \subseteq k^{\circ} \cdot FR \cdot k.$$

□

This succinct (allegorical) definition of a bisimulation is, we believe, original to this paper. In the following lemma, we show that it is nevertheless equivalent to the (categorical) definition proposed by Aczel and Mendler [3]. (Note that our definition generalises Aczel and Mendler's in two ways: the introduction of G, and the extension of dialgebras from maps to arbitrary relations. The lemma assumes, as they do, that the dialgebra is a map.)

Lemma 34. Suppose the map $k : FA \leftarrow GA$ is an (F,G)-dialgebra. Suppose also that $(f : A \leftarrow C,\ g : A \leftarrow C)$ is a tabulation of relation $R : A \leftarrow A$. Then R is a bisimulation of k if and only if there is a map $h : FC \leftarrow GC$ such that

$$k \cdot Gf = Ff \cdot h \wedge k \cdot Gg = Fg \cdot h.$$

Proof. The proof is by mutual implication. For the implication, assume that R is a bisimulation of k. We try to calculate a candidate for h:

$$
\begin{array}{ll}
 & k \cdot Gf = Ff \cdot h \ \wedge \ k \cdot Gg = Fg \cdot h \\
\equiv & \quad \{ \quad \text{all components are maps, shunting} \} \\
 & Ff^\circ \cdot k \cdot Gf \supseteq h \wedge Fg^\circ \cdot k \cdot Gg \supseteq h \\
\equiv & \quad \{ \quad \text{intersection} \} \\
 & Ff^\circ \cdot k \cdot Gf \cap Fg^\circ \cdot k \cdot Gg \ \supseteq \ h \ .
\end{array}
$$

Thus, using axiom (29), h exists if the relation $Ff^\circ \cdot \mathsf{out} \cdot Gf \cap Fg^\circ \cdot \mathsf{out} \cdot Gg$ is total, *i.e.* its right domain is id_{GC}. This we prove as follows:

$$
\begin{array}{ll}
 & id_{GC} \subseteq (Ff^\circ \cdot k \cdot Gf \cap Fg^\circ \cdot k \cdot Gg)> \\
\equiv & \quad \{ \quad id \subseteq (P \cap Q)> \ \equiv \ id \subseteq P^\circ \cdot Q \} \\
 & id_{GC} \subseteq Gf^\circ \cdot k^\circ \cdot Ff \cdot Fg^\circ \cdot k \cdot Gg \\
\equiv & \quad \{ \quad \text{shunting and converse, } F \text{ and } G \text{ are relators} \} \\
 & G(f \cdot g^\circ) \subseteq k^\circ \cdot F(f \cdot g^\circ) \cdot k \\
\equiv & \quad \{ \quad f \cdot g^\circ = R \} \\
 & GR \subseteq k^\circ \cdot FR \cdot k \\
\equiv & \quad \{ \quad \text{Def. 33} \} \\
 & R \text{ is a bisimulation of } k \ .
\end{array}
$$

Note that the above argument uses axiom (29). However, we do not need this axiom if we show that $Ff^\circ \cdot \mathsf{out} \cdot Gf \cap Fg^\circ \cdot \mathsf{out} \cdot Gg$ is itself a map. We have already shown that it is total so it remains to show that it is simple. This fact is proved assuming that the relator F preserves binary intersections in the following calculation:

$$
\begin{array}{ll}
 & (Ff^\circ \cdot \mathsf{out} \cdot Gf \cap Fg^\circ \cdot \mathsf{out} \cdot Gg) \cdot (Ff^\circ \cdot \mathsf{out} \cdot Gf \cap Fg^\circ \cdot \mathsf{out} \cdot Gg)^\circ \\
\subseteq & \quad \{ \quad \text{converse, monotonicity} \} \\
 & Ff^\circ \cdot \mathsf{out} \cdot Gf \cdot Gf^\circ \cdot \mathsf{out}^\circ \cdot Ff \ \cap \ Fg^\circ \cdot \mathsf{out} \cdot Gg \cdot Gg^\circ \cdot \mathsf{out}^\circ \cdot Fg \\
\subseteq & \quad \{ \quad \mathsf{out},\ Gf,\ Gg \text{ are all simple} \} \\
 & Ff^\circ \cdot Ff \ \cap \ Fg^\circ \cdot Fg \\
= & \quad \{ \quad F \text{ preserves binary intersections} \} \\
 & F(f^\circ \cdot f \cap g^\circ \cdot g) \\
= & \quad \{ \quad (f\,,g) \text{ is a tabulation: thus, } f^\circ \cdot f \cap g^\circ \cdot g \ = \ id_C \} \\
 & F id_C \ .
\end{array}
$$

For the follows-from, we have to show that if such a map h exists then R is a bisimulation of k. This is where we use the assumption that k is a map.

$$
\begin{array}{ll}
& R \text{ is a bisimulation of } k \\
\equiv & \qquad \{ \quad \text{definition of bisimulation } \} \\
& GR \subseteq k^\circ \cdot FR \cdot k \\
\equiv & \qquad \{ \quad R = f \cdot g^\circ, \ F \text{ and } G \text{ are relators } \} \\
& Gf \cdot Gg^\circ \subseteq k^\circ \cdot Ff \cdot Fg^\circ \cdot k \\
\equiv & \qquad \{ \quad k \text{ is a map, shunting } \} \\
& k \cdot Gf \cdot Gg^\circ \subseteq Ff \cdot Fg^\circ \cdot k \\
\Leftarrow & \qquad \{ \quad k \cdot Gf = Ff \cdot h, \text{ monotonicity } \} \\
& h \cdot Gg^\circ \subseteq Fg^\circ \cdot k \\
\equiv & \qquad \{ \quad \text{shunting } \} \\
& Fg \cdot h \subseteq k \cdot Gg \\
\equiv & \qquad \{ \quad \text{assumption } \} \\
& \text{true} \ .
\end{array}
$$

□

Instantiating R to **out** in (31) and (32), and comparing with Definition 33, we see that $[\![\mathsf{out}]\!]$ is the largest bisimulation of **out**. That is, by the computation rule (31), $[\![\mathsf{out}]\!]$ is a bisimulation of **out** and, by the coinduction rule (32), all bisimulations X of **out** satisfy $X \subseteq [\![\mathsf{out}]\!]$. Thus the next lemma has the well-known corollary that every bisimulation of a final coalgebra is at most the identity relation on the carrier set.

Lemma 35. $[\![\mathsf{out}]\!] = id_T$.

Proof. The inclusion $[\![\mathsf{out}]\!] \supseteq id_T$ follows from the fact that id_T is a bisimulation of out:

$$
\begin{array}{ll}
& id_T \subseteq [\![\mathsf{out}]\!] \\
\Leftarrow & \qquad \{ \quad (32) \ \} \\
& G id_T \subseteq \mathsf{out}^\circ \cdot F id_T \cdot \mathsf{out} \\
\equiv & \qquad \{ \quad F \text{ and } G \text{ are functors, } \mathbf{out} \text{ is total } \} \\
& \text{true} \ .
\end{array}
$$

So we are left with $[\![\mathsf{out}]\!] \subseteq id_T$. Let (f, g) be a tabulation of $[\![\mathsf{out}]\!]$. Then we have to prove that $f \cdot g^\circ \subseteq id_T$, or equivalently, $f = g$. Note that both f and g have the type $T \leftarrow C$ for some C. This suggests that we should use the unique extension property for dianamorphisms to prove the equality of f and g. The key to doing so is Lemma 34. In fact, we prove the stronger result that if R is any bisimulation of out and (f , g) is a tabulation of R then f and g are both equal to a dianamorphism whose existence is guaranteed by Lemma 34. Specifically, we have:

$$
\begin{array}{ll}
 & f \cdot g^\circ \subseteq id_T \\
\equiv & \quad \{ \quad \text{f and g are maps, shunting} \} \\
 & f = g \\
\Leftarrow & \quad \{ \quad \text{calculus} \} \\
 & \exists(h : \text{map } h :\ f = [\![h]\!] = g) \\
\equiv & \quad \{ \quad \text{unique extension property: Def. 2 (see also (3))} \} \\
 & \exists(h : \text{map } h :\ out \cdot Gf = Ff \cdot h \ \wedge\ out \cdot Gg = Fg \cdot h) \\
\equiv & \quad \{ \quad \text{(f , g) is a tabulation of R, Lem. 34} \} \\
 & \text{R is a bisimulation of out.}
\end{array}
$$

□

Now we can show that $[\_]$ is an extension of $[\![\_]\!]$. That is,

Lemma 36. For all maps f, $[\![f]\!] = [f]$.

Proof. The inclusion $\subseteq$ follows from coinduction:

$$
\begin{array}{ll}
 & [\![f]\!] \subseteq [f] \\
\Leftarrow & \quad \{ \quad \text{definition } [f]\text{, specifically (32)} \} \\
 & G[\![f]\!] \subseteq out^\circ \cdot F[\![f]\!] \cdot f \\
\equiv & \quad \{ \quad \text{all components are maps, shunting} \} \\
 & out \cdot G[\![f]\!] = F[\![f]\!] \cdot f \\
\equiv & \quad \{ \quad \text{unique extension property: Def. 2 (see also (3))} \} \\
 & true\ .
\end{array}
$$

It follows from this inclusion that [f] is total. So equality follows if we show that [f] is simple. The proof of this fact is the same as given in Lemma 20:

$$
\begin{array}{ll}
 & [f] \cdot [f]^\circ \subseteq id_T \\
\equiv & \quad \{ \quad \text{Lem. 35: } id_T = [out] \} \\
 & [f] \cdot [f]^\circ \subseteq [out] \\
\Leftarrow & \quad \{ \quad \text{coinduction: (32)} \} \\
 & G([f] \cdot [f]^\circ) \subseteq out^\circ \cdot F([f] \cdot [f]^\circ) \cdot out \\
\equiv & \quad \{ \quad \text{F and G are relators} \} \\
 & G[f] \cdot G[f]^\circ \subseteq out^\circ \cdot F[f] \cdot F[f]^\circ \cdot out
\end{array}
$$

$$
\begin{array}{ll}
\Leftarrow & \{ \quad \text{computation rule: (31),} \\
& \quad\ \ \text{monotonicity, and properties of converse} \} \\
& \mathsf{out}^\circ \cdot \mathsf{F}[\mathsf{f}] \cdot \mathsf{f} \cdot \mathsf{f}^\circ \cdot \mathsf{F}[\mathsf{f}]^\circ \cdot \mathsf{out} \subseteq \mathsf{out}^\circ \cdot \mathsf{F}[\mathsf{f}] \cdot \mathsf{F}[\mathsf{f}]^\circ \cdot \mathsf{out} \\
\equiv & \{ \quad \mathsf{f} \text{ is simple} \} \\
& \mathsf{true}\ .
\end{array}
$$

□

Now, using axiom (29) we can prove that the mapping $[\_]$ preserves total relations. The following obvious lemma is needed first, as well as later on.

Lemma 37. The mapping $[\_]$ is monotonic.

Proof. Straightforward combination of (31, 32) and monotonicity of composition. □

Lemma 38. The mapping $[\_]$ preserves total relations.

Proof. From axiom (29) it follows that for total relation R, there exists a map f such that $\mathsf{f} \subseteq \mathsf{R}$. Thus,

$$
\begin{array}{ll}
& [\mathsf{R}] \\
\supseteq & \{ \quad \mathsf{R} \supseteq \mathsf{f}, \text{ Lem. 37: } [\_] \text{ is monotonic} \} \\
& [\mathsf{f}] \\
= & \{ \quad \text{Lem. 36: } [\_] \text{ is an extension of } [\![\_]\!]. \} \\
& [\![\mathsf{f}]\!]\ .
\end{array}
$$

Hence $[\mathsf{R}]$ contains a total map. So $[\mathsf{R}]$ is total. □

Finally, we prove the cancellation law for the mapping $[\_]$:

Lemma 39. $[\mathsf{R}]^\circ \cdot [\mathsf{S}]$ is the largest solution of the equation

$$\mathsf{X} :: \ \mathsf{G}\mathsf{X} \subseteq \mathsf{R}^\circ \cdot \mathsf{F}\mathsf{X} \cdot \mathsf{S}.$$

Proof. It is easy to see that $[\mathsf{R}]^\circ \cdot [\mathsf{S}]$ is a solution of the given equation, since

$$
\begin{array}{ll}
& \mathsf{G}([\mathsf{R}]^\circ \cdot [\mathsf{S}]) \subseteq \mathsf{R}^\circ \cdot \mathsf{F}([\mathsf{R}]^\circ \cdot [\mathsf{S}]) \cdot \mathsf{S} \\
\Leftarrow & \{ \quad \mathsf{F} \text{ and } \mathsf{G} \text{ relators, computation} \} \\
& \mathsf{R}^\circ \cdot \mathsf{F}[\mathsf{R}]^\circ \cdot \mathsf{out} \cdot \mathsf{out}^\circ \cdot \mathsf{F}[\mathsf{S}] \cdot \mathsf{S} \subseteq \mathsf{R}^\circ \cdot \mathsf{F}[\mathsf{R}]^\circ \cdot \mathsf{F}[\mathsf{S}] \cdot \mathsf{S} \\
\equiv & \{ \quad \mathsf{out} \text{ simple} \} \\
& \mathsf{true}\ .
\end{array}
$$

It remains to prove that it is at least any other solution.

Let (f, g) be a tabulation of some solution of the given equation. Then we have to prove that $f \cdot g^{\circ} \subseteq [R]^{\circ} \cdot [S]$. We calculate:

$$
\begin{array}{ll}
& f \cdot g^{\circ} \subseteq [R]^{\circ} \cdot [S] \\
\equiv & \quad \{ \quad \text{shunting} \} \\
& id_C \subseteq f^{\circ} \cdot [R]^{\circ} \cdot [S] \cdot g \\
\Leftarrow & \quad \{ \quad \text{claim: } [Fh^{\circ} \cdot P \cdot Gh] \subseteq [P] \cdot h \} \\
& id_C \subseteq [Ff^{\circ} \cdot R \cdot Gf]^{\circ} \cdot [Fg^{\circ} \cdot S \cdot Gg] \\
\equiv & \quad \{ \quad id \subseteq P^{\circ} \cdot Q \equiv id \subseteq (P \cap Q)_{>} \} \\
& id_C \subseteq ([Ff^{\circ} \cdot R \cdot Gf] \cap [Fg^{\circ} \cdot S \cdot Gg])_{>} \\
\Leftarrow & \quad \{ \quad \text{by monotonicity Lem. 37: } [P \cap Q] \subseteq [P] \cap [Q] \} \\
& id_C \subseteq [Ff^{\circ} \cdot R \cdot Gf \cap Fg^{\circ} \cdot S \cdot Gg]_{>} \\
\Leftarrow & \quad \{ \quad \text{Lem. 38: } [\_] \text{ preserves total relations} \} \\
& id_C \subseteq (Ff^{\circ} \cdot R \cdot Gf \cap Fg^{\circ} \cdot S \cdot Gg)_{>} \\
\equiv & \quad \{ \quad id \subseteq (P \cap Q)_{>} \equiv id \subseteq P^{\circ} \cdot Q \} \\
& id_C \subseteq (Ff^{\circ} \cdot R \cdot Gf)^{\circ} \cdot Fg^{\circ} \cdot S \cdot Gg \\
\equiv & \quad \{ \quad \text{converse, shunting} \} \\
& Gf \cdot Gg^{\circ} \subseteq R^{\circ} \cdot Ff \cdot Fg^{\circ} \cdot S \\
\equiv & \quad \{ \quad F \text{ and } G \text{ relators} \} \\
& G(f \cdot g^{\circ}) \subseteq R^{\circ} \cdot F(f \cdot g^{\circ}) \cdot S \\
\equiv & \quad \{ \quad (f, g) \text{ is a tabulation of a solution of the given equation.} \\
& \text{Thus, } f \cdot g^{\circ} \text{ satisfies the equation.} \} \\
& \text{true.}
\end{array}
$$

The claim in the second step we verify by:

$$
\begin{array}{ll}
& [Fh^{\circ} \cdot P \cdot Gh] \subseteq [P] \cdot h \\
\equiv & \quad \{ \quad \text{shunting} \} \\
& [Fh^{\circ} \cdot P \cdot Gh] \cdot h^{\circ} \subseteq [P] \\
\Leftarrow & \quad \{ \quad \text{coinduction: (32)} \} \\
& G([Fh^{\circ} \cdot P \cdot Gh] \cdot h^{\circ}) \subseteq out^{\circ} \cdot F([Fh^{\circ} \cdot P \cdot Gh] \cdot h^{\circ}) \cdot P \\
\equiv & \quad \{ \quad F \text{ and } G \text{ relators, shunting} \} \\
& G[Fh^{\circ} \cdot P \cdot Gh] \subseteq out^{\circ} \cdot F[Fh^{\circ} \cdot P \cdot Gh] \cdot Fh^{\circ} \cdot P \cdot Gh
\end{array}
$$

$$\equiv \qquad \{ \quad \text{computation: (31)} \}$$

$\mathsf{true}.$

□

This concludes the proof of the main theorem.

7. Conclusion

This paper outlines the beginning of an attempt to smoothline the development of generic programming. We have proposed a unifying definition of the coregular relators which encompasses both the coinductive relators like stream and the non-coinductive relators. In addition we have proved that a dialgebra is final in the underlying map category if and only if it is a relational final dialgebra in the allegory itself.

There is much that needs to be done. Problems that are of particular interest to us are whether it is possible to establish in one go that all regular relators have membership [12] and that any pair of (co)regular relators commute [10, 11] rather than resort to the cumbersome case analyses as we have done in the past.

We are grateful to the anonymous referee for several insightful suggestions.

References

[1] C.J. Aarts, R.C. Backhouse, P. Hoogendijk, T.S. Voermans and J. van der Woude, A relational theory of datatypes. Available *via* World-Wide Web at `http://www.win.tue.nl/cs/wp/papers` (September 1992).

[2] P. Aczel, *Non Well-Founded Sets*, Number 14 in CSLI Lecture Notes. Center for the Study of Language and Information (Stanford, California, 1988).

[3] P. Aczel and N. Mendler, A final coalgebra theorem, D.H. Pitt, Ed., *Category Theory and Computer Science*. Springer Verlag, *Lecture Notes in Comput. Sci.* (1989) 357–365.

[4] R.C. Backhouse, P. de Bruin, P. Hoogendijk, G. Malcolm, T.S. Voermans and J. van der Woude, Polynomial relators, M. Nivat, C.S. Rattray, T. Rus and G. Scollo, Eds., in *Proc. of the 2nd Conference on Algebraic Methodology and Software Technology, AMAST'91*. Springer-Verlag, Workshops in Computing (1992) 303–326.

[5] R. Bird, O. de Moor and P. Hoogendijk, Generic functional programming with types and relations. *J. Funct. Programming* **6** (1996) 1–28.

[6] R.S. Bird and O. de Moor, *Algebra of Programming.* Prentice-Hall International (1996).

[7] H. Doornbos, *Reductivity arguments and program construction.* Ph.D. Thesis, Eindhoven University of Technology, Department of Mathematics and Computing Science (1996).

[8] P.J. Freyd and A. Ščedrov, *Categories, Allegories.* North-Holland (1990).

[9] T. Hagino, A typed lambda calculus with categorical type constructors, D.H. Pitt, A. Poigne and D.E. Rydeheard, Eds., *Category Theory and Computer Science.* Springer-Verlag, *Lecture Notes in Comput. Sci.* **283** (1988) 140–57.

[10] P. Hoogendijk, *A Generic Theory of Datatypes.* Ph.D. Thesis, Department of Mathematics and Computing Science, Eindhoven University of Technology (1997).

[11] P. Hoogendijk and R. Backhouse, When do datatypes commute? E. Moggi and G. Rosolini, Eds., *Category Theory and Computer Science, 7th International Conference.* Springer-Verlag, *Lecture Notes in Comput. Sci.* **1290** (1997) 242–260.

[12] P. Hoogendijk and O. de Moor, What is a datatype? Technical Report 96/16, Department of Mathematics and Computing Science, Eindhoven University of Technology, 1996. *J. Funct. Programming*, to appear.

[13] B. Jacobs and J. Rutten, A tutorial on (co)algebras and (co)induction. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS* **62** (1997) 222–259.

[14] P. Jansson and J. Jeuring, PolyP - a polytypic programming language extension. In *POPL '97: The 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages.* ACM Press (1997) 470–482.

[15] C.B. Jay, A semantics for shape. *Sci. Comput. Programming* **25** (1995) 251–283.

[16] C.B. Jay and J.R.B. Cockett, Shapely types and shape polymorphism, D. Sannella, Ed., *ESOP '94: 5th European Symposium on Programming.* Springer Verlag, *Lecture Notes in Comput. Sci.* (1994) 302–316.

[17] J. Jeuring, Polytypic pattern matching. In *Conference Record of FPCA '95, SIGPLAN-SIGARCH-WG2.8 Conference on Functional Programming Languages and Computer Architecture* (1995) 238–248.

[18] J. Jeuring and P. Jansson, Polytypic programming, J. Launchbury, E. Meijer and T. Sheard, Eds., *Proceedings of the Second International Summer School on Advanced Functional Programming Techniques.* Springer-Verlag, *Lecture Notes in Comput. Sci.* **1129** (1996) 68–114.

[19] J. Lambek, A fixpoint theorem for complete categories. *Math. Z.* **103** (1968) 151–161.

[20] J. Lambek, Subequalizers. *Canad. Math. Bull.* **13** (1970) 337–349.

[21] S. MacLane, *Categories for the Working Mathematician.* Springer-Verlag, New York (1971).

[22] L. Meertens, Calculate polytypically! H. Kuchen and S. Doaitse Swierstra, Eds., *Proceedings of the Eighth International Symposium PLILP '96 Programming Languages: Implementations, Logics and Programs.* Springer Verlag, *Lecture Notes in Comput. Sci.* **1140** (1996) 1–16.

[23] E. Meijer, M.M. Fokkinga and R. Paterson, Functional programming with bananas, lenses, envelopes and barbed wire. In *FPCA91: Functional Programming Languages and Computer Architecture.* Springer-Verlag, *Lecture Notes in Comput. Sci.* **523** (1991) 124–144.

[24] S.D. Swierstra and O. de Moor, Virtual data structures, H. Partsch, B. Möller and S. Schuman, Eds., *Formal Program Development.* Springer-Verlag, *Lecture Notes in Comput. Sci.* **755** (1993) 355–371.

Received October 27, 1998. Revised April 29, 1999.