Categories of Containers

Michael Abbott[1], Thorsten Altenkirch[2], and Neil Ghani[1]

[1] Department of Mathematics and Computer Science, University of Leicester
michael@araneidae.co.uk,ng13@mcs.le.ac.uk

[2] School of Computer Science and Information Technology, Nottingham University
txa@cs.nott.ac.uk

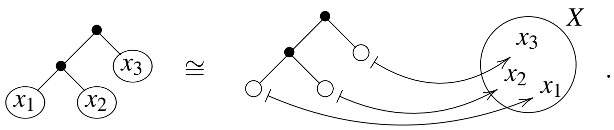
Abstract. We introduce the notion of containers as a mathematical formalisation of the idea that many important datatypes consist of templates where data is stored. We show that containers have good closure properties under a variety of constructions including the formation of initial algebras and final coalgebras. We also show that containers include strictly positive types and shapely types but that there are containers which do not correspond to either of these. Further, we derive a representation result classifying the nature of polymorphic functions between containers. We finish this paper with an application to the theory of shapely types and refer to a forthcoming paper which applies this theory to differentiable types.

1 Introduction

Any element of the type $\mathsf{List}(X)$ of lists over X can be uniquely written as a natural number n given by the length of the list, together with a function $\{1,\ldots,n\} \to X$ which labels each position within the list with an element from X:

$$n:\mathbb{N} \ , \qquad \sigma:\{1\,..\,n\} \to X \ .$$

Similarly, any binary tree tree can be described by its underlying shape which is obtained by deleting the data stored at the leaves with a function mapping the positions in this shape to the data thus:



More generally, we are led to consider datatypes which are given by a set of shapes S and, for each $s \in S$, a family of positions $P(s)$. This presentation of the datatype defines an endofunctor $X \mapsto \coprod_{s\in S} X^{P(s)}$ on **Set**. In this paper we formalise these intuitions by considering families of objects in a locally cartesian closed category $\mathbb{C}$, where the family $s:S \vdash P(s)$ is represented by an object $P \in \mathbb{C}/S$, and the associated functor $T_{S\triangleright P}:\mathbb{C} \to \mathbb{C}$ is defined by $T_{S\triangleright P}X \equiv \Sigma s:S.\,(P(s) \Rightarrow X)$.

We begin by constructing a category $\mathscr{G}$ of "containers", ie syntactic presentations of shapes and positions, and define a full and faithful functor T to the category of endofunctors of $\mathbb{C}$. Given that polymorphic functions are natural transformations, full and faithfulness allows us to classify polymorphic functions between container functors in terms of their action on the shapes and positions of the underlying containers.

We show that $\mathscr{G}$ is complete and cocomplete and that limits and coproducts are preserved by T. This immediately shows that i) container types are closed under products, coproducts and subset types; and ii) this semantics is compositional in that the semantics of a datatype is constructed canonically from the semantics of its parts. The construction of initial algebras and final coalgebras of containers requires, firstly, the definition of containers with multiple parameters and, secondly, a detailed analysis of when T preserves limits and colimits of certain filtered diagrams.

We conclude the paper by relating containers to the shapely types of Jay and Cockett (1994) and Jay (1995). The definition of shapely types does not require the hypothesis of local cartesian closure which we assume, but when $\mathbb{C}$ *is* locally cartesian closed then it turns out that the shapely types are precisely the functors generated by the "discretely finite" containers. A container is discretely finite precisely when each of its objects of positions is locally isomorphic to a finite cardinal.

Further, we gain much by the introduction of extra categorical structure, e.g. the ability to form initial algebras and final coalgebras of containers and the representation result concerning natural transformations between containers. Unlike containers, shapely types are *not* closed under the construction of coinductive types, since the position object of an infinite list cannot be discretely finite.

In this paper we assume that $\mathbb{C}$ is locally finitely presentable (lfp), hence complete and cocomplete, which excludes several interesting examples including Scott domains and realisability models. Here we use the lfp structure for the construction of initial algebras and final coalgebras. In future work we expect to replace this assumption with a more delicate treatment of induction using *internal* structure.

Another application of containers is as a foundation for generic programming within a dependently typed programming framework (Altenkirch and McBride, 2002). An instance of this theme, the derivatives of functors as suggested in McBride (2001), is developed in Abbott et al. (2003) using the material presented here.

The use of the word *container* to refer to a class of datatypes can be found in Hoogendijk and de Moor (2000) who investigated them in a relational setting: their containers are actually closed under quotienting. Containers as introduced here are closely related to analytical functors, which were introduced by Joyal, see Hasegawa (2002). Here we consider them in a more general setting by looking at locally cartesian categories with some additional properties. In the case of **Set** containers are a generalisation of normal functors, closing them under quotients would generalise analytical functors.

In summary, this paper makes the following contributions:

- We develop a new and generic concept of what a container is which is applicable to a wide range of semantic domains.
- We give a representation theorem (Theorem 3.4) which provides a simple analysis of polymorphic functions between datatypes.
- We show a number of closure properties of the category of containers which allow us to interpret all strictly positive types by containers.

– We lay the foundation for a theory of generic programming; a first application is the theory of differentiable datatypes as presented in Abbott et al. (2003).
– We show that Jay and Cockett's shapely types are all containers.

2 Definitions and Notation

This paper implicitly uses the machinery of fibrations (Jacobs 1999, Borceux 1994, chapter 8, etc) to develop the key properties of container categories, and in particular the fullness of the functor T relies on the use of fibred natural transformations. This section collects together the key definitions and results required in this paper.

Given a category with finite limits $\mathbb{C}$, refer to the slice category $\mathbb{C}/A$ over $A \in \mathbb{C}$ as the *fibre* of $\mathbb{C}$ over A. Pullbacks in $\mathbb{C}$ allow us to lift each $f : A \to B$ in $\mathbb{C}$ to a *pullback* or *reindexing* functor $f^* : \mathbb{C}/B \to \mathbb{C}/A$. Assigning a fibre category to each object of $\mathbb{C}$ and a reindexing functor to each morphism of $\mathbb{C}$ is (subject to certain coherence equations) a presentation of a *fibration over* $\mathbb{C}$.

Composition with f yields a functor $\Sigma_f : \mathbb{C}/A \to \mathbb{C}/B$ left adjoint to f^*. $\mathbb{C}$ is *locally cartesian closed* iff each fibre of $\mathbb{C}$ is cartesian closed, or equivalently, if each pullback functor f^* has a right adjoint $f^* \dashv \Pi_f$.

Each exponential category $\mathbb{C}^I$ can in turn be regarded as fibred over $\mathbb{C}$ by taking the fibre of $\mathbb{C}^I$ over $A \in \mathbb{C}$ equal to $(\mathbb{C}/A)^I$. Now define $[\mathbb{C}^I, \mathbb{C}^J]$ to be the category of fibred functors $F : \mathbb{C}^I \to \mathbb{C}^J$ and fibred natural transformations, where each F is a family $F_A : (\mathbb{C}/A)^I \to (\mathbb{C}/A)^J$ such that $(f^*)^J F_B \cong F_A (f^*)^I$ for each $f : A \to B$ and similarly for natural transformations.

Write $a : A \vdash B(a)$ or even just $A \vdash B$ for $B \in \mathbb{C}/A$. We'll write $a : A, b : B(a) \vdash C(a,b)$ as a shorthand for $(a,b) : \Sigma_A B \vdash C(a,b)$. When dealing with a collection A_i for $i \in I$, we'll write this as $(A_i)_{i \in I}$ or $\vec{A}$ or even just A. Write $\Sigma a : A$ and $\Pi a : A$ for the Σ and Π types corresponding to the adjoints to reindexing. Substitution in variables will be used interchangeably with substitution by pullback, so $A \vdash f^* B$ may also be written as $a : A \vdash B(f(a))$ or $a : A \vdash B(fa)$. The signs $\coprod$ and $\prod$ will be used for coproducts and products respectively over external sets, while Σ and Π refer to the corresponding internal constructions in $\mathbb{C}$. See Hofmann (1997) for a more detailed explanation of the interaction between type theory and semantics assumed in this paper.

Limits and colimits are *fibred* iff they exist in each fibre and are preserved by reindexing functors. Limits and colimits in a locally cartesian closed category $\mathbb{C}$ are automatically fibred. This useful result allows us to omit the qualification that limits and colimits be "fibred" throughout this paper.

When $\mathbb{C}$ is locally cartesian closed say that coproducts are *disjoint* (or equivalently that $\mathbb{C}$ is *extensive*)[1] iff the pullback of distinct coprojections $\kappa_i : A_i \to \coprod_{i \in I} A_i$ into a coproduct is always the initial object 0. Henceforth, we'll assume that $\mathbb{C}$ has finite limits, is locally cartesian closed and has disjoint coproducts. The following notion of "disjoint fibres" follows from disjoint coproducts.

[1] For general $\mathbb{C}$, coproducts are disjoint iff coprojections are also mono, and $\mathbb{C}$ is extensive iff coproducts are disjoint and are preserved by pullbacks.

Proposition 2.1. *If $\mathbb{C}$ has disjoint coproducts then the functor $\vec{\kappa}^* : \mathbb{C}/\coprod_{i\in I} A_i \to \prod_{i\in I}(\mathbb{C}/A_i)$, taking $\coprod_{i\in I} A_i \vdash B$ to $(A_i \vdash \kappa_i^* B)_{i\in I}$, is an equivalence. Say that $\mathbb{C}$ has* disjoint fibres *when this holds.* $\square$

Write $\overset{\circ}{\coprod} : \prod_{i\in I}(\mathbb{C}/A_i) \to \mathbb{C}/\coprod_{i\in I} A_i$ for the adjoint to $\vec{\kappa}^*$ and $-\overset{\circ}{+}-$ for the binary case. Note that $\overset{\circ}{\coprod}_{i\in I} B_i \cong \coprod_{i\in I} \Sigma_{\kappa_i} B_i$ for $(A_i \vdash B_i)_{i\in I} \in \prod_{i\in I}(\mathbb{C}/A_i)$.

The following lemma collects together some useful identities which hold in any category considered in this paper.

Lemma 2.2. *For extensive locally cartesian closed $\mathbb{C}$ the following isomorphisms hold (*IC *stands for* intensional choice*,* Cu *for* Curry *and* DF *for* disjoint fibres*):*

$$\Pi a\!:\!A.\,\Sigma b\!:\!B(a).\,C(a,b) \cong \Sigma f : (\Pi a\!:\!A.\,B(a)).\,\Pi a\!:\!A.\,C(a, fa) \tag{IC1}$$

$$\textstyle\prod_{i\in I} \Sigma b\!:\!B_i.\,C_i(b) \cong \Sigma a : \prod_{i\in I} B_i.\,\prod_{i\in I} C_i(\pi_i a) \tag{IC2}$$

$$\Pi a\!:\!A.\,(B(a) \Rightarrow C) \cong (\Sigma a\!:\!A.\,B(a)) \Rightarrow C \tag{Cu1}$$

$$\textstyle\prod_{i\in I}(B_i \Rightarrow C) \cong \left(\coprod_{i\in I} B_i\right) \Rightarrow C \tag{Cu2}$$

$$\left(\overset{\circ}{\coprod}_{i\in I} B_i\right)(\kappa_i a) \cong B_i(a) \tag{DF1}$$

$$\textstyle\coprod_{i\in I} \Sigma a\!:\!A_i.\,C(\kappa_i a) \cong \Sigma a : \coprod_{i\in I} A_i.\,C(a) \tag{DF2}$$

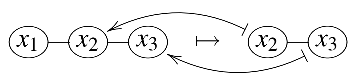
$\square$

For technical convenience, a choice of pullbacks is assumed in $\mathbb{C}$ (this ensures that our fibrations are cloven). Finally, note that we make essential use of classical set theory with choice in the meta-theory in theorem 5.6 and proposition 6.6. It should be possible to avoid this dependency by developing more of the theory internally to $\mathbb{C}$.

3 Basic Properties of Containers

The basic notion of a *container* is a dependent pair of types $A \vdash B$ creating a functor $T_{A\triangleright B}X \equiv \Sigma a\!:\!A.\,(B(a) \Rightarrow X)$. In order to understand a morphism of containers, consider the map $\mathsf{tail} : \mathsf{List}\,X \to 1 + \mathsf{List}\,X$ taking the empty list to 1 and otherwise yielding the tail of the given list:

tail of list



This map is defined by i) a choice of shape in $1 + \mathsf{List}\,X$ for each shape in $\mathsf{List}\,X$; and ii) for each position in the chosen shape a position in the original shape. Thus a morphism of containers $(A \vdash B) \to (C \vdash D)$ is a pair of morphisms $(u\!:\!A \to C,\, f\!:\!u^*D \to B)$. With this definition of a category $\mathscr{G}$ of containers we can construct a full and faithful functor $T : \mathscr{G} \to [\mathbb{C}, \mathbb{C}]$ and show the completeness properties discussed in the introduction.

However, when constructing fixed points it is also necessary to take account of containers with parameters, so we define $T : \mathscr{G}_I \to [\mathbb{C}^I, \mathbb{C}]$ for each parameter index set I. For the purposes of this paper the index set n or I will generally be a finite set, but this makes little difference. Indeed, it is straightforward to generalise the development in this paper to the case where containers are parameterised by *internal* index objects $I \in \mathbb{C}$; when $\mathbb{C}$ has enough coproducts nothing is lost by doing this, since $\mathbb{C}^I \simeq \mathbb{C}/\coprod_{i\in I} 1$. This generalisation will be important for future developments of this theory, but is not required in this paper.

Definition 3.1. *Given an index set I define the* category of containers $\mathcal{G}_I$ *as follows:*

- *Objects are pairs $(A \in \mathbb{C}, B \in (\mathbb{C}/A)^I)$; write this as $(A \triangleright B) \in \mathcal{G}_I$*
- *A morphism $(A \triangleright B) \to (C \triangleright D)$ is a pair (u, f) for $u : A \to C$ in $\mathbb{C}$ and $f : (u^*)^I D \to B$ in $(\mathbb{C}/A)^I$.*

A container $(A \triangleright B) \in \mathcal{G}_I$ can be written using type theoretic notation as

$$\vdash A \qquad i:I, a:A \vdash B_i(a) \enspace .$$

A morphism $(u, f) : (A \triangleright B) \to (C \triangleright D)$ can be written in type theoretic notation as

$$u : A \to C \qquad i:I, a:A \vdash f_i(a) : D_i(ua) \to B_i(a) \enspace .$$

Finally, each $(A \triangleright B) \in \mathcal{G}_I$, thought of as a syntactic presentation of a datatype, generates a fibred functor $T_{A\triangleright B} : \mathbb{C}^I \to \mathbb{C}$ which is its semantics.

Definition 3.2. *Define the* container construction functor $T : \mathcal{G}_I \to [\mathbb{C}^I, \mathbb{C}]$ *as follows. Given $(A \triangleright B) \in \mathcal{G}_I$ and $X \in \mathbb{C}^I$ define*

$$T_{A\triangleright B} X \equiv \Sigma a : A. \prod_{i\in I} (B_i(a) \Rightarrow X_i) \enspace ,$$

and for $(u, f) : (A \triangleright B) \to (C \triangleright D)$ define $T_{u,f} : T_{A\triangleright B} \to T_{C\triangleright D}$ to be the natural transformation $T_{u,f} X : T_{A\triangleright B} X \to T_{C\triangleright D} X$ thus:

$$(a, g) : T_{A\triangleright B} X \vdash T_{u,f} X(a, g) \equiv (u(a), (g_i \cdot f_i)_{i\in I}) \enspace .$$

The following proposition follows more or less immediately by the construction of T.

Proposition 3.3. *For each container $F \in \mathcal{G}_I$ and each container morphism $\alpha : F \to G$ the functor T_F and natural transformation T_α are fibred over $\mathbb{C}$.* $\square$

By making essential use of the fact that the natural transformations in $[\mathbb{C}^I, \mathbb{C}]$ are *fibred* (c.f. section 2) we can show that T is full and faithful.

Theorem 3.4. *The functor $T : \mathcal{G}_I \to [\mathbb{C}^I, \mathbb{C}]$ is full and faithful.*

Proof. To show that T is full and faithful it is sufficient to lift each natural transformation $\alpha : T_{A\triangleright B} \to T_{C\triangleright D}$ in $[\mathbb{C}^I, \mathbb{C}]$ to a map $(u_\alpha, f_\alpha) : (A \triangleright B) \to (C \triangleright D)$ in $\mathcal{G}_I$ and show this construction is inverse to T.

Given $\alpha : T_{A\triangleright B} \to T_{C\triangleright D}$ construct $\ell \equiv (a', \mathrm{id}_{B(a')}) \in T_{A\triangleright B} B$ in the fibre $\mathbb{C}/A$ (or in terms of type theory, add $a' : A$ to the context). We can now construct $\alpha B \cdot \ell \in T_{C\triangleright D} B = \Sigma c : C. \prod_{i\in I} (D_i(c) \Rightarrow B_i(a'))$ in the same context, and write $\alpha B \cdot \ell = (u_\alpha, f_\alpha)$ where $u_\alpha(a') : C$ and $f_\alpha(a') : \prod_{i\in I} (D_i(u_\alpha a') \Rightarrow B_i(a'))$ for $a' : A$.

Thus (u_α, f_α) can be understood as a morphism $(A \triangleright B) \to (C \triangleright D)$ in $\mathscr{G}_I$. It remains to show that this construction is inverse to T.

When $\alpha = T_{u,f}$, just evaluate $\alpha B \cdot \ell = (ua', \mathrm{id} \cdot f)$, which corresponds to the original map (u, f).

To show in general that $\alpha = T_{u_\alpha, f_\alpha}$, let $X \in \mathbb{C}^I$, $a : A$ and $g : \prod_{i \in I}(B_i(a) \Rightarrow X_i)$ be given, consider the diagram

$$\begin{tikzcd}
1 \arrow[r, "\ell"] \arrow[dr, "{(u_\alpha a, f_\alpha(a))}"'] & T_{A \triangleright B} B \arrow[r, "T_{A \triangleright B} g"] \arrow[d, "\alpha B"] & T_{A \triangleright B} X \arrow[d, "\alpha X"] \\
& T_{C \triangleright D} B \arrow[r, "T_{C \triangleright D} g"'] & T_{C \triangleright D} X
\end{tikzcd}$$

and evaluate

$$\begin{aligned}
\alpha X \cdot (a, g) &= \alpha X \cdot T_{A \triangleright B} g \cdot \ell = T_{C \triangleright D} g \cdot \alpha B \cdot \ell = T_{C \triangleright D} g \cdot (u_\alpha a, f_\alpha(a)) \\
&= (u_\alpha a, g \cdot f_\alpha(a)) = T_{u_\alpha, f_\alpha} X \cdot (a, g) \quad .
\end{aligned}$$

This shows that $\alpha = T_{u_\alpha, f_\alpha}$ as required. □

This theorem gives a particularly simple analysis of polymorphic functions between container functors. For example, it is easy to observe that there are precisely n^m polymorphic functions $X^n \to X^m$: the data type X^n is the container $(1 \triangleright n)$ and hence there is a bijection between polymorphic functions $X^n \to X^m$ and functions $m \to n$. Similarly, any polymorphic function $\mathsf{List}\, X \to \mathsf{List}\, X$ can be uniquely written as a function $u : \mathbb{N} \to \mathbb{N}$ together with for each natural number $n : \mathbb{N}$ a function $f_n : un \to n$.

4 Limits and Colimits of Containers

It turns out that each $\mathscr{G}_I$ inherits completeness and cocompleteness from $\mathbb{C}$, and that T preserves completeness. Preservation of cocompleteness is more complex, and only a limited class of colimits are preserved by T.

Proposition 4.1. *If $\mathbb{C}$ has limits and colimits of shape $\mathbb{J}$ then $\mathscr{G}_I$ has limits of shape $\mathbb{J}$ and T preserves these limits.*

Proof. We'll proceed by appealing to the fact that T reflects limits (since it is full and faithful), and the proof will proceed separately for products and equalisers.

Products. Let $(A_k \triangleright B_k)_{k \in K}$ be a family of objects in $\mathscr{G}_I$ and compute (the labels refer to lemma 2.2)

$$\begin{aligned}
\prod\nolimits_{k \in K} T_{A_k \triangleright B_k} X &= \prod\nolimits_{k \in K} \Sigma a : A. \prod\nolimits_{i \in I} (B_{k,i}(a) \Rightarrow X_i) \\
&\cong \Sigma a : \prod\nolimits_{k \in K} A_k. \prod\nolimits_{k \in K} \prod\nolimits_{i \in I} (B_{k,i}(\pi_k a) \Rightarrow X_i) && \text{(IC2)} \\
&\cong \Sigma a : \prod\nolimits_{k \in K} A_k. \prod\nolimits_{i \in I} \Big(\big(\coprod\nolimits_{k \in K} B_{k,i}(\pi_k a) \big) \Rightarrow X_i \Big) && \text{(Cu2)} \\
&= T_{\prod_{k \in K} A_k \triangleright \coprod_{k \in K} (\pi_k^*)^I B_k} X
\end{aligned}$$

showing by reflection along T that

$$\prod_{k\in K}(A_k \rhd B_k) \cong \Big(\prod_{k\in K} A_k \ \rhd\ \coprod_{k\in K}(\pi_k^*)^I B_k\Big) \quad .$$

Equalisers. Given parallel maps $(u,f),(v,g):(A \rhd B) \rightrightarrows (C \rhd D)$ construct

$$\begin{tikzcd}
(E \rhd Q) \arrow[r, tail, "{(e,q)}"] & (A \rhd B) \arrow[r, shift left, "{(u,f)}"] \arrow[r, shift right, "{(v,g)}"'] & (C \rhd D)
\end{tikzcd}$$

where e is the equaliser in $\mathbb{C}$ of u, v and q is the coequaliser in $(\mathbb{C}/E)^I$ of $(e^*)^I f$, $(e^*)^I g$. To show that $T_{e,q}$ is the equaliser of $T_{u,f}, T_{v,g}$ fix $X \in \mathbb{C}^I$, $U \in \mathbb{C}$ and let $\alpha : U \to T_{A\rhd B}X$ be given equalising this parallel pair at X.

For $x:U$ write $\alpha(x) = (a,h)$ where $a:A$, $h:\prod_{i\in I}(B_i(a) \Rightarrow X_i)$. The condition on α tells us that $u(a) = v(a)$ and so there is a unique $y:E$ with $a = e(y)$. Similarly we know that $h\cdot f(ey) = h\cdot g(ey)$ and in particular there is a unique $k:Q(y) \to X$ with $h = k\cdot q$.

The assignment $x \mapsto (y,k)$ defines a map $\beta : U \to T_{E\rhd Q}X$ giving a unique factorisation of α, showing that $T_{e,q}X$ is an equaliser and hence so is (e,q). $\square$

In particular, this result tells us that the limit in $[\mathbb{C}^I, \mathbb{C}]$ of a diagram of container functors is itself a container functor.

It's nice to see that coproducts of containers are also well behaved.

Proposition 4.2. *If $\mathbb{C}$ has products and coproducts of size K then $\mathscr{G}_I$ has coproducts of size K preserved by T.*

Proof. Given a family $(A_k \vdash B_k)_{k\in K}$ of objects in $\mathscr{G}_I$ calculate (making essential use of disjointness of fibres):

$$\begin{aligned}
\coprod_{k\in K} T_{A_k\rhd B_k}X &= \coprod_{k\in K} \Sigma a : A_k.\prod_{i\in I}(B_{k,i}(a) \Rightarrow X_i) \\
&\cong \coprod_{k\in K} \Sigma a : A_k.\prod_{i\in I}\Big(\big(\biguplus_{k'\in K} B_{k',i}\big)(\kappa_k a) \Rightarrow X_i\Big) && \text{(DF1)} \\
&\cong \Sigma a : \coprod_{k\in K} A_k.\prod_{i\in I}\Big(\big(\biguplus_{k\in K} B_{k,i}\big)(a) \Rightarrow X_i\Big) && \text{(DF2)} \\
&= T_{\coprod_{k\in K} A_k \rhd (\biguplus_{k\in K} B_{j,i})_{i\in I}} X
\end{aligned}$$

showing by reflection along T that

$$\coprod_{k\in K}(A_k \rhd B_k) \cong \Big(\coprod_{k\in K} A_k \ \rhd\ \biguplus_{k\in K} B_k\Big) \quad . \qquad \square$$

The fate of coequalisers is more complicated. It turns out that $\mathscr{G}_I$ has coequalisers when $\mathbb{C}$ has both equalisers and coequalisers, but they are *not* preserved by T.

The following proposition is presented without proof (the construction of coequalisers in $\mathscr{G}$ is fairly complex and is not required in this paper).

Proposition 4.3. *If $\mathbb{C}$ has equalisers and coequalisers then $\mathscr{G}_I$ has coequalisers.* $\square$

The following example shows that coequalisers are not preserved by T.

Example 4.4. Consider the following coequaliser diagram in $[\mathbb{C},\mathbb{C}]$

$$\begin{tikzcd} X\times X \arrow[r, shift left, "\mathrm{id}_{X\times X}"] \arrow[r, shift right, "(\pi',\pi)"'] & X\times X \arrow[r, two heads] & (X\times X)/\sim \end{tikzcd}$$

where $(x,y)\sim(y,x)$. The functor $X\mapsto X\times X$ is a container functor generated by $(1\triangleright 2)$, and the coequaliser of the corresponding parallel pair in $\mathscr{G}_1$ is the container $(1\triangleright 0)$. Note however that $T_{1\triangleright 0}X\cong 1\ncong(X\times X)/\sim$.

Unfortunately, filtered colimits aren't preserved by T either.

Example 4.5. Consider the ω-chain in $\mathscr{G}_1$ given by $n\mapsto(1\triangleright A^n)$ (for fixed A) on objects and $(n\to n+m)\mapsto\pi_{n,m}:A^{n+m}\cong A^n\times A^m\to A^n$ on maps. The filtered colimit of this diagram can be computed in $\mathscr{G}_1$ to be $(1\triangleright A^{\mathbb{N}})$. However, applying T to this diagram produces the ω-chain

$$X\xrightarrow{X^{\pi_{0,1}}}X^A\xrightarrow{X^{\pi_{1,1}}}X^{A^2}\xrightarrow{X^{\pi_{2,1}}}\cdots$$

and the colimit of this chain in **Set** is strictly smaller than $X^{A^{\mathbb{N}}}$.

5 Filtered Colimits of Cartesian Diagrams

Although $\mathscr{G}_I$ has colimits they are not preserved by T, and this also applies to filtered colimits. As we will want to use filtered colimits for the construction of initial algebras, this is a potential problem. Fortunately, there exists a class of filtered colimit diagrams which is both sufficient for the construction of initial algebras and which *are* preserved by T.

Throughout this section take $\mathbb{C}$ to be finitely accessible ($\mathbb{C}$ has filtered colimits and a generating set of finitely presentable objects, Adámek and Rosický 1994) as well as being locally cartesian closed.

Definition 5.1. *A morphism (u,f) in $\mathscr{G}_I$ is* cartesian *iff f is an isomorphism*[2].

For each u there is a bijection between cartesian morphisms $(u,f):(A\triangleright B)\to(C\triangleright D)$ in $\mathscr{G}_I$ and morphisms $\bar{f}$ in $\mathbb{C}^I$ making each square below a pullback:

$$\begin{tikzcd} B_i \arrow[r, "\bar{f}_i"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & D_i \arrow[d] \\ A \arrow[r, "u"'] & C \end{tikzcd} .$$

[2] (u,f) is *cartesian* with respect to this definition precisely when it is cartesian (in the sense of fibrations) with respect to the projection functor $\pi:\mathscr{G}_I\to\mathbb{C}$ taking $(A\triangleright B)$ to A.

We can also translate the notion of cartesian morphism into natural transformations between container functors: a natural transformation $\alpha : T_{A\triangleright B} \to T_{C\triangleright D}$ derives from a cartesian map iff the naturality squares of α are all pullbacks (such natural transformations are often also called *cartesian*, in this case with respect to the "evaluation at 1" functor).

Define $\widehat{\mathscr{G}_I}$ to have the same objects as $\mathscr{G}_I$ but only cartesian arrows as morphisms. We will show that $\widehat{\mathscr{G}_I}$ has filtered colimits which are preserved by T (when restricted to $\widehat{\mathscr{G}_I}$), and hence also by the inclusion $\widehat{\mathscr{G}_I} \hookrightarrow \mathscr{G}_I$.

The lemma below follows directly from the corresponding result in **Set** and helps us work with maps from finitely presentable objects to filtered colimits (write $\bigvee D$ for the colimit of a filtered diagram D).

Lemma 5.2. *Let $D : \mathbb{J} \to \mathbb{C}$ be a filtered diagram with colimiting cone $d : D \to \bigvee D$ and let U be finitely presentable.*

1. *For each $\alpha : U \to \bigvee D$ there exists $J \in \mathbb{J}$ and $\alpha_J : U \to DJ$ such that $\alpha = d_J \cdot \alpha_J$.*
2. *Given $\alpha : U \to DI$, $\beta : U \to DJ$ such that $d_I \cdot \alpha = d_J \cdot \beta$ there exists $K \in \mathbb{J}$ and maps $f : I \to K$, $g : J \to K$ such that $Df \cdot \alpha = Dg \cdot \beta$.* □

Before the main result we need a technical lemma about filtered colimits in finitely accessible categories.

Lemma 5.3. *Given a filtered diagram in $\mathbb{C}^{\to}$ with every edge a pullback then the arrows of the colimiting cone are also pullbacks.*

Proof. We need to show, for each $I \in \mathbb{C}$, that the square

$$\begin{tikzcd}
EI \arrow[r, "e_I"] \arrow[d, "\alpha_I"'] & \bigvee E \arrow[d, "\bar{\alpha}"] \\
DI \arrow[r, "d_I"'] & \bigvee D
\end{tikzcd}$$

is a pullback, where $E \xrightarrow{\alpha} D$ is the diagram, (d, e) are the components of its colimiting cone and $\bar{\alpha}$ is the factorisation of $d \cdot \alpha$ through e. So let a cone $DI \xleftarrow{a} U \xrightarrow{b} \bigvee E$ satisfying $d_I \cdot a = \bar{\alpha} \cdot b$ be given. Without loss of generality we can assume that U is finitely presentable and we can now appeal to lemma 5.2 above.

Construct first $b_J : U \to EJ$ such that $b = e_J \cdot b_J$; then as $d_I \cdot a = \bar{\alpha} \cdot e_J \cdot b_J = d_J \cdot (\alpha_J \cdot b_J)$ there exist $f : I \to K$, $g : J \to K$ with $Df \cdot a = Dg \cdot \alpha_J \cdot b_J = \alpha_K \cdot Eg \cdot b_J$ and so we can construct a factorisation $b_I : U \to EI$ through the pullback over f satisfying $\alpha_I \cdot b_I = a$ and $Ef \cdot b_I = Eg \cdot b_J$. This is a factorisation of (a, b) since $e_I \cdot b_I = e_K \cdot Ef \cdot b_I = e_K \cdot Eg \cdot e_J = e_J \cdot b_J = b$.

This factorisation is unique. Let $b, b' : U \rightrightarrows EI$ be given such that $e_I \cdot b = e_I \cdot b'$. Then there exist $f, f' : I \rightrightarrows J$ with $Ef \cdot b = Ef' \cdot b'$; but indeed there exists $g : J \to K$ with $h \equiv g \cdot f = g \cdot f'$ and so $Eh \cdot b = Eh \cdot b'$. As the square over h is a pullback we can conclude $b = b'$. □

Now we are in a position to state the main result, that the filtered colimit of a cartesian diagram of container functors is itself a container functor.

Proposition 5.4. *For each set I the category $\widehat{\mathscr{G}_I}$ has filtered colimits which are preserved by T.*

Proof. Let a diagram $(D \triangleright E) : \mathbb{J} \to \widehat{\mathscr{G}_I}$ be given, i.e. for each $K \in \mathbb{J}$ there is a container $(DK \triangleright EK)$ and for each $f : K \to L$ a cartesian container morphism (Df, Ef).

For each $f : K \to L$ in $\mathbb{J}$, write $\bar{E}f$ for the map $EK \to EL$ derived from cartesian Ef so that we get the left hand pullback square below:

$$\begin{tikzcd}
EK \arrow[r, "\bar{E}f"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & EL \arrow[r, "\bar{e}_L"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \bigvee \bar{E} \arrow[d] \\
DK \arrow[r, "\bar{D}f"'] & DL \arrow[r, "d_L"'] & \bigvee D
\end{tikzcd} .$$

After taking the colimits shown (with colimiting cones d and $\bar{e}$), we know from lemma 5.3 that the right hand square is also a pullback and we can interpret the right hand side as a container together with a cartesian cone $(d,e) : (D \triangleright E) \dot{\to} (\bigvee D \triangleright \bigvee \bar{E})$.

It remains to show that $T_{\bigvee D \triangleright \bigvee \bar{E}} \cong \bigvee T_{D \triangleright E}$, so let a cone $f : T_{D \triangleright E} X \dot{\to} U$ be given as shown below, where the map k_K takes (a, g) to $(d_K(a), g)$, using the isomorphism $(\bigvee \bar{E})_i(d_K(a)) \cong EK_i(a)$ (for $K \in \mathbb{J}$, $i : I$, $a : DK_j$) derived from (d,e) cartesian.

$$\begin{tikzcd}
\Sigma a : DK_j . \prod_{i \in I}(EK_i(a) \Rightarrow X_i) \arrow[r, "k_K"] \arrow[dr, "f_K"'] & \Sigma a : \bigvee D . \prod_{i \in I}((\bigvee \bar{E})_i(a) \Rightarrow X_i) \arrow[d, dashed, "h"] \\
& U
\end{tikzcd}$$

To construct h let $a : \bigvee D$ and $g : \prod_{i \in i}((\bigvee D)_i(a) \Rightarrow X_i)$ be given and choose $K \in \mathbb{J}$, $a_K \in DK$ such that $a = d_K(a_K)$, and so we have $(a_K, g) : T_{DK \triangleright EK} X$ and can compute $h(a,g) \equiv f_K(a_K, g)$; this construction of $h(a,g)$ is unique and independent of the choice of K and a_K. $\square$

Finally the above proposition can be applied to the construction of fixed points on $\mathscr{G}_I$.

Definition 5.5. *Say that an endofunctor F on a category with filtered colimits* has rank *iff there exists a cardinal $\aleph$, the* rank *of F, such that F preserves $\aleph$-filtered colimits.*

The following theorem is a variant of Adámek and Koubek (1979).

Theorem 5.6 (Adámek). *If a category $\mathbb{C}$ has an initial object and colimits of all filtered diagrams then every endofunctor on $\mathbb{C}$ with rank has an initial algebra.*

If $G : \mathbb{C} \to \mathbb{D}$ preserves the initial object and all filtered colimits then any endofunctor F' on $\mathbb{D}$ satisfying $F'G \cong GF$ for some endofunctor F on $\mathbb{C}$ with rank has an initial algebra given by the image under G of the initial algebra of F. $\square$

The construction of initial algebras in $\mathscr{G}$ now follows as a corollary of the above.

Theorem 5.7. *Let F be an endofunctor on $\mathscr{G}_I$ such that F restricts to an endofunctor $\hat{F}$ on $\widehat{\mathscr{G}_I}$ (i.e., F preserves cartesian morphisms) and such that $\hat{F}$ has rank, then F has an initial algebra $\mu F \in \mathscr{G}_I$ which is preserved by T.*

Proof. We've established that $\widehat{\mathscr{G}_I}$ has filtered colimits which are preserved by $\widehat{\mathscr{G}_I} \hookrightarrow \mathscr{G}_I$ and by T and it's clear that the initial object of $\mathscr{G}_I$ is initial in $\widehat{\mathscr{G}_I}$ and is also preserved by T and so we can apply theorem 5.6. $\square$

As noted in section 2 it would be desirable to have a constructive version of this theorem, probably along the lines suggested by Taylor (1999, Section 6.7).

6 Fixed Points of Containers

Categories of containers are, under suitable assumptions, closed under the operations of taking least and greatest fixed points, or in other words given a container functor $F(\vec{X},Y)$ in $n+1$ parameters the types $\mu Y.F(\vec{X},Y)$ and $\nu Y.F(\vec{X},Y)$ are containers (in n parameters).

The least and greatest fixed points of a type are defined by repeated substitution, for example the type $\nu Y.F(\vec{X},Y)$ can be constructed as the limit of the ω-chain

$$1 \longleftarrow F(\vec{X},1) \longleftarrow F(\vec{X},F(\vec{X},1)) \longleftarrow \cdots \longleftarrow \varprojlim_{n<\omega} F^n[1]$$

where we write $F[Y] \equiv F(\vec{X},Y)$ (note that the ν type only needs ω-limits for its construction, but as discussed below, μ types can require colimits of transfinite chains[3]). Therefore the first thing we need to do is to define the composition of two containers.

Given containers $F \in \mathscr{G}_{I+1}$ and $G \in \mathscr{G}_I$ we can compose their images under T to construct the functor

$$T_F[T_G] \equiv (\mathbb{C}^I \xrightarrow{(\mathrm{id}_{\mathbb{C}^I}, T_G)} \mathbb{C}^I \times \mathbb{C} \cong \mathbb{C}^{I+1} \xrightarrow{T_F} \mathbb{C}) \ .$$

This composition can be lifted to a functor $-[-] : \mathscr{G}_{I+1} \times \mathscr{G}_I \to \mathscr{G}_I$ as follows. For a container in $\mathscr{G}_{I+1}$ write $(A \triangleright B, E) \in \mathscr{G}_{I+1}$, where $B \in (\mathbb{C}/A)^I$ and $E \in \mathbb{C}/A$ and define:

$$(A \triangleright B,E)[(C \triangleright D)] \equiv (a:A, f:E(a) \Rightarrow C \ \triangleright\ (B_i(a) + \Sigma e:E(a).D_i(fe))_{i\in I}) \ .$$

In other words, given type constructors $F(\vec{X},Y)$ and $G(\vec{X})$ this construction defines the composite type constructor $F[G](\vec{X}) \equiv F(\vec{X},G(\vec{X}))$.

[3] For example, the type of ω-branching trees, $\mu Y.X + (\mathbb{N} \Rightarrow Y)$, cannot be constructed using only ω-colimits.

Proposition 6.1. *Composition of containers commutes with composition of functors thus: $T_F[T_G] \cong T_{F[G]}$.*

Proof. Calculate (for conciseness we write exponentials using superscripts where convenient and write Σ_A for $\Sigma a : A.$ throughout, eliding the parameter a):

$$
\begin{aligned}
T_{A \triangleright B,E}[T_{C \triangleright D}]X &= \Sigma_A \Big(\big(\textstyle\prod_{i \in I} X_i^{B_i}\big) \times \big(E \Rightarrow \Sigma c : C. \textstyle\prod_{i \in I} X_i^{D_i(c)}\big) \Big) \\
&\cong \Sigma_A \Big(\big(\textstyle\prod_{i \in I} X_i^{B_i}\big) \times \big(\Sigma f : C^E. \Pi e : E. \textstyle\prod_{i \in I} X_i^{D_i(fe)}\big) \Big) && \text{(IC1)} \\
&\cong \Sigma_A \Sigma f : C^E. \textstyle\prod_{i \in I} \Big(X_i^{B_i} \times \big(\Pi e : E. X_i^{D_i(fe)}\big) \Big) \\
&\cong \Sigma_A \Sigma f : C^E. \textstyle\prod_{i \in I} \big((B_i + \Sigma e : E. D_i(fe)) \Rightarrow X_i \big) && \text{(Cu1, Cu2)} \\
&\cong T_{(A \triangleright B,E)[C \triangleright D]} X \quad .
\end{aligned}
$$

As all the above isomorphisms are natural in X we get the desired isomorphism of functors. ☐

The next lemma is useful for the construction of both least and greatest fixed points and has other applications. In particular, T_F preserves both pullbacks and cofiltered limits.

Lemma 6.2. *For $(A \triangleright B) \in \mathscr{G}_I$ the functor $T_{A \triangleright B}$ preserves limits of connected non-empty diagrams (*connected limits*).*

Proof. Since $\prod$ and $\Rightarrow$ preserve limits, it is sufficient to observe that Σ_A preserves connected limits, which is noted, for example, in Carboni and Johnstone (1995). ☐

Corollary 6.3. *For each $F \in \mathscr{G}_{I+1}$ the functor $F[-] : \mathscr{G}_I \to \mathscr{G}_I$ preserves connected limits.*

Proof. Let D be a non-empty connected diagram, then since T_F preserves connected limits it is easy to see that $T_F[\varprojlim D] \cong \varprojlim(T_F[D])$. Since T preserves limits we can calculate

$$T_{F[\varprojlim D]} \cong T_F[T_{\varprojlim D}] \cong T_F[\varprojlim T_D] \cong \varprojlim(T_F[T_D]) \cong \varprojlim T_{F[D]} \cong T_{\varprojlim(F[D])}$$

and so by reflection along T conclude that $F[\varprojlim D] \cong \varprojlim(F[D])$. ☐

We can immediately conclude that if $\mathbb{C}$ is complete and cocomplete (in fact, ω- limits and colimits are sufficient) then containers have final coalgebras.

Theorem 6.4. *Each $F \in \mathscr{G}_{I+1}$ has a final coalgebra $\nu F \in \mathscr{G}_I$ which is preserved by T (and so satisfies $T_{\nu F} \cong \nu T_F$).*

Proof. Since $F[-]$ preserves limits of ω-chains the final coalgebra of F can be constructed as the limit $\varprojlim_{n<\omega} F^n[1]$, and since T preserves this limit the fixed point is also preserved by T, by the dual of theorem 5.6 ☐

For the construction of least fixed points (or initial algebras) two more preliminary results are needed. First we need to show that the construction of the fixed point can be restricted to $\widehat{\mathscr{G}}$, so that we know that it will be preserved by T.

Proposition 6.5. *The functor* $-[-] : \mathcal{G}_{I+1} \times \mathcal{G}_I \to \mathcal{G}_I$ *restricts to a functor on the category of cartesian container morphisms,* $-[-] : \widehat{\mathcal{G}}_{I+1} \times \widehat{\mathcal{G}}_I \to \widehat{\mathcal{G}}_I$.

Proof. It is sufficient to show that when $\alpha : F \to F'$ and $\beta : G \to G'$ are both cartesian then so is $\alpha[\beta]$, and indeed it is sufficient to show that $T_\alpha[T_\beta]$ is a cartesian natural transformation. This follows immediately from the fact that T_F preserves pullbacks and that T_α and T_β are cartesian natural transformations. $\square$

Secondly we need to show that $F[-]$ has rank. Assume from now to the end of this section that $\mathbb{C}$ is a finitely[4] accessible category.

Proposition 6.6. *When* $\mathbb{C}$ *is finitely accessible, every container functor has rank.*

Proof. Let $(A \triangleright B) \in \mathcal{G}_I$ be a container. We first need to establish the result

$$\prod_{i\in I} \left(B_i \Rightarrow \bigvee_{j\in\mathbb{J}} X_{j,i}\right) \cong \bigvee_{j\in\mathbb{J}} \prod_{i\in I} (B_i \Rightarrow X_{j,i})$$

for sufficiently large $\aleph$ and $\aleph$-filtered $\mathbb{J}$, which we do by appealing to two results of Adámek and Rosický (1994). First, we know (from their theorem 2.39) that each functor category $\mathbb{C}^I$ is accessible, and secondly we know from their proposition 2.23 that each functor with an adjoint between accessible categories has rank.

Now since Σ_A preserves colimits we can conclude that $T_{A\triangleright B}$ has rank. $\square$

Corollary 6.7. *For each* $F \in \mathcal{G}_{I+1}$ *the endofunctor* $F[-]$ *on* $\mathcal{G}_I$ *restricts to an endofunctor on* $\widehat{\mathcal{G}}_I$ *with rank.*

Proof. Let $\aleph$ be the rank of T_F and let D be an $\aleph$-filtered diagram in $\widehat{\mathcal{G}}$. We know that $T_F[-]$ will preserve $\bigvee D$ so we can now repeat the calculation of corollary 6.3 to conclude that $F[-]$ also has rank $\aleph$. $\square$

That containers have least fixed points now follows from corollary 6.7 and theorem 5.7.

Theorem 6.8. *Each* $F \in \mathcal{G}_{I+1}$ *has a least fixed point* $\mu F \in \mathcal{G}_I$ *satisfying* $T_{\mu F} \cong \mu T_F$. $\square$

7 Strictly Positive Types

We now return to the point that all strictly positive types can be described as containers.

Definition 7.1. *A* strictly positive type in n variables *(Abel and Altenkirch, 2000) is a type expression (with type variables* $X_1, \ldots, X_n$*) built up according to the following rules:*

- *if* K *is a constant type (with no type variables) then* K *is a strictly positive type;*
- *each type variable* X_i *is a strictly positive type;*
- *if* U, V *are strictly positive types then so are* $U + V$ *and* $U \times V$*;*

[4] The qualification *finitely* is not strictly necessary here.

- *if K is a constant type and U a strictly positive type then $K \Rightarrow U$ is a strictly positive type;*
- *if U is a strictly positive type in $n+1$ variables then $\mu X.U$ and $\nu X.U$ are strictly positive types in n variables (for X any type variable).*

Note that the type expression for a strictly positive type U can be interpreted as a functor $U : \mathbb{C}^n \to \mathbb{C}$, and indeed we can see that each strictly positive type corresponds to a container in $\mathscr{G}_n$.

Let strictly positive types U, V be represented by containers $(A \rhd B)$ and $(C \rhd D)$ respectively, then the table below shows the correspondence between strictly positive types and containers[5].

$$\begin{aligned} K &\mapsto (K \rhd 0) & X_i &\mapsto (1 \rhd (\delta_{i,j})_{j \in I}) \\ U + V &\mapsto (A + C \rhd B \,\natural\, D) & U \times V &\mapsto (a : A, c : C \rhd B(a) \times D(c)) \\ K \Rightarrow U &\mapsto (f : K \Rightarrow A \rhd \Sigma k : K.B(fk)) \end{aligned}$$

The construction of fixed points is a bit more difficult to describe in type-theoretic terms. Let W be represented by $(A \rhd B, E) \in \mathscr{G}_I$ (see section 6), then for any fixed point C of $T_{A \rhd E}$ with $\Phi : T_{A \rhd E} C \cong C$ we can define $C \vdash D_C$ as the initial solution of

$$D_C(\Phi(a, f)) \cong B(a) + \Sigma e : E.D_C(fe) \;\; ; \tag{*}$$

we can now define

$$\begin{aligned} \mu X.W &\mapsto (\mu X.T_{A \rhd E} X \rhd D_{\mu X.T_{A \rhd E} X}) \\ \nu X.W &\mapsto (\nu X.T_{A \rhd E} X \rhd D_{\nu X.T_{A \rhd E} X}) \;\; . \end{aligned}$$

All the initial and terminal (co)algebras used above can be constructed explicitly using the results of section 6. It is interesting to note that μ and ν only differ in the type of shapes but that the type of positions can be defined uniformly.

Indeed, consider $F(X) = \mu Y.1 + X \times Y$, then $\mu X.F(X)$ is the type of lists and as we have already observed the type of shapes is isomorphic to $\mathbb{N} \cong \mu X.1 + X$ and the family of positions over n can be conveniently described by $P(n) = \{i \mid i < n\}$. Dually, $\nu X.F(X)$ is the type of lazy (i.e. potentially infinite) lists. The type of shapes is given by $\mathbb{N}^{\mathrm{co}} = \nu X.1 + X$, the *conatural numbers*, which contain a fixed point of the successor $\omega = \mathrm{s}(\omega) : \mathbb{N}^{\mathrm{co}}$. Hence $P(\omega) \cong \mathbb{N}$ and this represents the infinite lists whose elements can be indexed by the natural numbers. Had we used the *terminal* solution of (*) to construct the type of positions, then the representation of infinite lists would incorrectly have an additional *infinite* position.

In the reverse direction it seems that there are containers which do not correspond to strictly positive types. A probable counterexample is the type of nests, defined as the least solution to the equation

$$N(Y) \cong 1 + Y \times N(Y \times Y) \;\; .$$

The datatype N *is* a container since it can be written as $N(X) \cong \Sigma n : \mathbb{N}.X^{2^n - 1}$, but it should be possible to show that it is not strictly positive following the argument used in Moggi et al. (1999) to show that the type of square matrices is not regular.

[5] We write $\delta_{i,j} \equiv 1$ iff $i = j$ and $\delta_{i,j} \equiv 0$ otherwise.

8 Relationships with Shapely Types

In Jay and Cockett (1994) and Jay (1995) “shapely types” (in one parameter) in a category $\mathbb{C}$ are defined to be strong pullback preserving functors $\mathbb{C} \to \mathbb{C}$ equipped with a strong cartesian natural transformation to the list type, where the *list type* is the initial algebra $\mu Y.1 + X \times Y$.

To see the relationship with containers, note that proposition 2.6.11 of Jacobs (1999) tells us that strong pullback preserving functors are in bijection with fibred pullback preserving functors, and similarly strong natural transformations between such functors correspond to fibred natural transformations. The next proposition will allow us to immediately observe that shapely types are containers.

Proposition 8.1. *Any functor $G \in [\mathbb{C}^I, \mathbb{C}]$ equipped with a cartesian natural transformation $\alpha : G \to T_F$ to a container functor is itself isomorphic to a container functor.*

Proof. Let $F \equiv (A \rhd B)$ then $(\alpha_1, \mathrm{id}_{\alpha_1^* B}) : (G1 \rhd \alpha_1^* B) \to (A \rhd B)$ is a cartesian map in $\mathscr{G}_I$; this yields a cartesian natural transformation $T_{G1 \rhd \alpha_1^* B} \to T_{A \rhd B}$. It now follows from the observation that each α_X makes GX the pullback along α_1 of the map $T_{A \rhd B} X \to A$ that $G \cong T_{G1 \rhd \alpha_1^* B}$ as required. □

Since the “list type” is given by the container $(n : \mathbb{N} \rhd [n])$, it immediately follows (when $\mathbb{C}$ is locally cartesian closed) that every shapely type is a container functor.

In the opposite direction, containers which are locally isomorphic to finite cardinals give rise to shapely types. To see this, we follow Johnstone (1977) and refer to the object $[-] \in \mathbb{C}/\mathbb{N}$, which can be constructed as the morphism $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mapping $(n, m) \mapsto n + m + 1$, as the object of *finite cardinals* in $\mathbb{C}$.

Definition 8.2. *An object $A \vdash B$ is* discretely finite *iff there exists a morphism $u : A \to \mathbb{N}$ such that $B \cong u^*[-]$, i.e. each fibre $a : A \vdash B(a)$ is isomorphic to a finite cardinal.*

Say that a container $(A \rhd B) \in \mathscr{G}_I$ is discretely finite *iff each component B_i for $i \in I$ is discretely finite.*

Note that “discretely finite” is strictly stronger than finitely presentable and other possible notions of finiteness. An immediate consequence of this definition is that the object of finite cardinals is a *generic object* for the category of discretely finite containers, and the following theorem relating shapely types and containers now follows as a corollary.

Theorem 8.3. *In a locally cartesian closed category with a natural number object the category of shapely functors and strong natural transformations is equivalent to the category of discretely finite containers.* □

However, this paper tells us more about shapely types. In particular, containers show how to extend shapely types to cover coinductive types. Finally, the representation result for containers clearly translates into a representation result classifying the polymorphic functions between shapely types.

It interesting to note that the “traversals” of Moggi et al. (1999) do not carry over to containers in general, for example the type $\mathbb{N} \Rightarrow X$ does not effectively traverse over the lifting monad $X \mapsto X + 1$.

References

M. Abbott, T. Altenkirch, N. Ghani, and C. McBride. Derivatives of containers. URL http://www.cs.nott.ac.uk/˜txa/. Submitted for publication, 2003.

A. Abel and T. Altenkirch. A predicative strong normalisation proof for a λ-calculus with interleaving inductive types. In *Types for Proof and Programs,TYPES '99*, volume 1956 of *Lecture Notes in Computer Science*, 2000.

J. Adámek and V. Koubek. Least fixed point of a functor. *Journal of Computer and System Sciences*, 19:163–178, 1979.

J. Adámek and J. Rosický. *Locally Presentable and Accessible Categories*. Number 189 in London Mathematical Society Lecture Note Series. Cambridge University Press, 1994.

T. Altenkirch and C. McBride. Generic programming within dependently typed programming. In *IFIP Working Conference on Generic Programming*, 2002.

F. Borceux. *Handbook of Categorical Algebra 2*. Cambridge University Press, 1994.

A. Carboni and P. Johnstone. Connected limits, familial representability and Artin glueing. *Math. Struct. in Comp. Science*, 5:441–459, 1995.

R. Hasegawa. Two applications of analytic functors. *Theoretical Computer Science*, 272(1-2):112–175, 2002.

M. Hofmann. Syntax and semantics of dependent types. In A. M. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation*, volume 14, pages 79–130. Cambridge University Press, Cambridge, 1997.

P. Hoogendijk and O. de Moor. Container types categorically. *Journal of Functional Programming*, 10(2):191–225, 2000.

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

C. B. Jay. A semantics for shape. *Science of Computer Programming*, 25:251–283, 1995.

C. B. Jay and J. R. B. Cockett. Shapely types and shape polymorphism. In *ESOP '94: 5th European Symposium on Programming*, Lecture Notes in Computer Science, pages 302–316. Springer-Verlag, 1994.

P. T. Johnstone. *Topos Theory*. Academic Press, 1977.

C. McBride. The derivative of a regular type is its type of one-hole contexts. URL http://www.dur.ac.uk/c.t.mcbride/. 2001.

E. Moggi, G. Bellè, and C. B. Jay. Monads, shapely functors and traversals. *Electronic Notes in Theoretical Computer Science*, 29, 1999.

P. Taylor. *Practical Foundations of Mathematics*. Cambridge University Press, 1999.