Derivatives of Containers

Michael Abbott[1], Thorsten Altenkirch[2], Neil Ghani[1], and Conor McBride[3]

[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
[3] Department of Computer Science, University of Durham
c.t.mcbride@durham.ac.uk

Abstract. We are investigating McBride's idea that the type of one-hole contexts are the formal derivative of a functor from a categorical perspective. Exploiting our recent work on containers we are able to characterise derivatives by a universal property and show that the laws of calculus including a rule for initial algebras as presented by McBride hold — hence the differentiable containers include those generated by polynomials and least fixpoints. Finally, we discuss abstract containers (i.e. quotients of containers) — this includes a container which plays the role of e^x in calculus by being its own derivative.

1 Introduction

In his classic functional pearl Huet (1997) shows how to represent a tree with one of its subtrees 'in focus' by a pair of the subtree and the one-hole context (or 'zipper') in which it sits. The unpublished article McBride (2001) gives a 'generic program' for computing the type of one-hole contexts for any regular inductive datatype: remarkably, the key step is to *differentiate* the functor which generates the datatype by the rules we learned from Leibniz (1684). It was an observation in search of an explanation. In this paper, we find such an explanation from a categorical perspective.

Our categorical presentation of *containers* in Abbott, Altenkirch, and Ghani (2003) provides the key to the mystery. We now specify differentiation by a universal property in the category of containers, more precisely $\partial F \cong \mathrm{Id} \multimap F$ where $H \multimap -$ is the right adjoint of $- \times H$ in the category $\widehat{\mathscr{G}}$ of *cartesian* morphisms between containers. We thus uncover the linear notion of *tangent* which McBride's programs mechanise.

Given this specification by universal property, we verify Leibniz's laws, McBride's extension for initial algebras (via the chain rule) and further conjecture an extension to terminal coalgebras. We note also that there are also containers which are not differentiable, such as $(\mathbb{N} \Rightarrow \mathbb{N}) \Rightarrow -$.

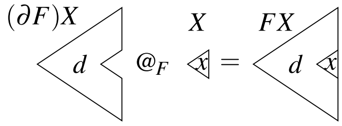
Our present work rediscovers the notion of a derivative of an analytic functor, Joyal (1986), from a computational perspective. We also generalise his approach, considering a more general class of functors and categories. Inspired by Joyal, we introduce the notion of an *abstract container*, which arises when closing containers under coequalisers. In particular we seek out the container which corresponds to the exponential function: just like e^x, it is its own derivative.

Acknowledgements

Our thanks to the anonymous referees who provided helpful insights, and to Peter Hancock for many interesting discussions.

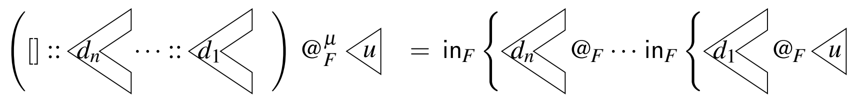
2 Background

McBride's article gives the computational intuition to differentiating analytic functors. He presents *syntactically* a class of functors F, closed under polynomial operations and (first-order) least fixed point. He then shows that (for a general X), $(\partial F)X$ represents 'an FX with one hole for an X', and gives the corresponding *linear application* ('plugging-in') operator, $@_F$ in the diagram below:



Curried, $@_F$ embeds $(\partial F)X$ into the function space $X \to FX$, of which it is the linear fragment. One immediate application is to make Huet's notion of 'zipper' generic over inductive datatypes μF wherever ∂F can be defined. The single constructor $\mathsf{in}_F \;:\; F(\mu F) \to \mu F$ making an element of μF from a 'container' of its immediate subelements, is an isomorphism. Hence, $(\partial F)(\mu F)$ gives 'elements of μF with a hole for one immediate sub-μF'. Just as the 'subterm' relation is the reflexive-transitive closure of the 'immediate subterm' relation, we can represent 'elements of μF with a hole for one sub-μF' by *lists*, $\mathsf{List}(\partial F)(\mu F))$, where each element is a step on the path from hole to root, recording the surrounding data. Here, lists grow 'on the right', echoing Huet's account of 'zippers' as *stacks*. The corresponding application operation iterates $@_F$:

$$@^{\mu}_{F} \;:\; \mathsf{List}((\partial F)(\mu F)) \times \mu F \to \mu F$$



The zipper data structure has many uses: Huet's example is 'structure editing', with $\mathsf{List}((\partial F)(\mu F)) \times \mu F$ exactly representing μF with a 'cursor' at one node. Many tree-rewriting operations can thus be expressed in terms of more primitive navigating (moving the cursor) and editing (replacing the term at the cursor) steps. Context-sensitive operations need merely inspect the list.

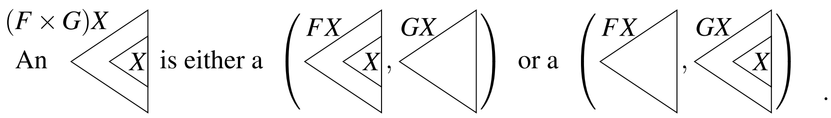
Moreover, we gain a language with which we can express general properties of datatypes more easily, and we gain tools with which to prove them. For example, 'inductive datatypes have no cycles' becomes

$$ds \mathbin{@^{\mu}_{F}} u = u \;\Rightarrow\; ds = [] \;\; .$$

Generically, structural induction on μF amounts to showing $P(\mathsf{in}_F\, t)$ given hypotheses $P(u)$ for each d, u such that $t = d \mathbin{@_F} u$.

2.1 What is ∂?

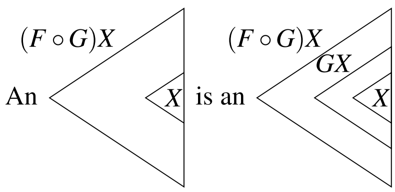
How can we explain ∂F in terms of F? Intuitively, $d \mathbin{@_F} x$ computes an element of FX, using x exactly once, and without inspecting x. That is, as Huet (2003) suggests, ∂F corresponds to $X \multimap FX$, the *linear* fragment of $X \to FX$, in a sense which we make precise in Definition 4.2. We can think of this linear approximation, $X \multimap FX$, as a kind of *tangent* to F at X. Similarly, Ehrhard and Regnier (2001) use differentiation to compute linear approximants to λ-terms. Hence it is not so remarkable that ∂ behaves like differentiation—let us check some familiar laws informally. Consider an element of $(F \times G)$, containing a particular chosen element of X.



$$(\partial(F \times G))X \quad \cong \quad (\partial F)X \quad \times \quad GX \quad + \quad FX \quad \times \quad (\partial G)X$$

The X is either in the FX part or the GX part. Now, $(\partial(F \times G))X$ must record the possible contexts for our chosen X — the 'chevrons' which remain when the X is cut out. If the X lies within the FX, we must record a $(\partial F)X$ and the whole of the GX, and correspondingly in the other case. Hence the law holds.

Next, choosing an X within an $(F \circ G)X$ amounts to choosing a GX within an $F(GX)$, then an X within the GX.



$$(\partial(F \circ G))X \cong (\partial F)(GX) \times (\partial G)X$$

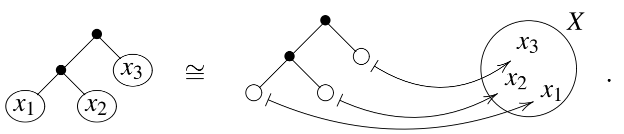
The X's context comes in two parts, corresponding with Leibniz's 'chain rule'.

2.2 Containers and Derivatives

Containers are a formalisation of the idea that many important datatypes consist of templates where data is stored. For example, any element of the type of lists $\mathsf{List}\ X$ can be uniquely written as a natural number n, the length of the list, together with a function $\{1,\ldots,n\} \to X$ which labels each position in the list with an element from X:

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

Similarly, any binary tree is given by its underlying shape (obtained by deleting the data stored at the leaves) and a mapping from positions in this shape to the data thus:



Thus 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 $(P_s)_{s \in S}$. This presentation of the datatype defines an endofunctor $X \mapsto \coprod_{s \in S} X^{P_s}$ on **Set**. More generally, Abbott et al. (2003) formalises 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 or *container* $T_{S \triangleright P} : \mathbb{C} \to \mathbb{C}$ is defined by

$$T_{S \triangleright P} X \equiv \Sigma s : S.\, (P(s) \Rightarrow X) \quad .$$

In effect, this gives a generalised 'power series' for functors representing containers of X. We might hope that their derivatives might obey a law like

$$\partial T_{S \triangleright P} X \cong \Sigma s : S.\, (P(s) \times ((P(s) - 1) \Rightarrow X))$$

and in fact they do, provided we can explain what we mean by $(P(s) - 1)$. Deleting an X within some $(s, f) : T_{S \triangleright P} X$ amounts to choosing the $p : P(s)$ which points to it. In doing so, we effectively select the shape of the context for the X; then such a context contains an X for each position in $P(s)$ *except* p. The analogue of $P(s) - 1$ is thus

$$\Sigma p' : P(s).\, \neg \mathrm{Eq}(p, p')$$

where p is the position of the missing X. For this to behave properly, $P(s)$ must possess a *decidable equality*, so that we can tell if any p' is the hole position, p:

$$\mathrm{Eq}(p, p') + \neg \mathrm{Eq}(p, p') \cong 1$$

so, indeed, for each $p : P(s)$:

$$P(s) \cong \Sigma p' : P(s).\, \neg \mathrm{Eq}(p, p') + 1 \quad .$$

2.3 Notation and Conventions

We use the same general semantic domain as in Abbott et al. (2003) where the category $\mathbb{C}$ is extensive[1], locally cartesian closed and locally finitely presentable (Adámek and Rosický, 1994).

We write $a : A \vdash B(a)$ or just $A \vdash B$ for $B \in \mathbb{C}/A$, and similarly $\Sigma a : A.B(a)$ and $\Sigma_A B$ for the domain of B regarded as an object. We'll write $a : A, b : B(a) \vdash C(a, b)$ as a shorthand for $(a, b) : \Sigma_A B \vdash C(a, b)$. We omit variables and weakenings when the meaning is clear from the context. For example, given $A \vdash B$ we can interchangeably write this as $a : A, c : C \vdash B(a)$ or $A \times C \vdash \pi^* B$ (for $\pi : A \times C \to A$).

For $A \vdash B$ write $\pi_B : \Sigma_A B \to A$ for the map projecting out the first component and write $\pi'_B : 1 \to \pi_B^* B$ for the morphism in $\mathbb{C}/\Sigma_A B$ corresponding to the second variable $a : A, b : B(a) \vdash b : B(a)$. Here we follow Hofmann (1994). When A and B are objects of $\mathbb{C}$ we may interchangeably write $A^* B$ or $\pi_A^* B$ for the object $\pi : A \times B \to A$ of $\mathbb{C}/A$.

[1] In general a category $\mathbb{C}$ with coproducts is *extensive* iff coproducts are preserved by pullbacks, coprojections into coproducts are monomorphisms, and 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. In a cartesian closed category all conditions except the last automatically hold.

Given $A \vdash B$ and $u : C \to A$ write $u_B : \Sigma_C u^* B \to \Sigma_A B$ for the map making the following square a pullback

$$\begin{tikzcd}
\Sigma_C u^* B \arrow[r, "u_B"] \arrow[d, "\pi_{u^*B}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Sigma_A B \arrow[d, "\pi_B"] \\
C \arrow[r, "u"'] & A
\end{tikzcd}$$

We'll write $\neg B \equiv 0^B$. We'll interchangeably write $B \Rightarrow X$ and X^B for the local exponent. Given an object $B \in \mathbb{C}$ the equality type $B \times B \vdash \mathrm{Eq}_B$ (or Eq when the type B is obvious from context) is given by the morphism $\delta : B \to B \times B$.

We denote the *global coproduct* of $A \vdash B$ and $C \vdash D$ by $A + C \vdash B \mathbin{\overset{\circ}{+}} D$. We denote the fibred coproduct of objects $A \vdash B$ and $A \vdash D$ over a common base by $A \vdash B + D$.

3 The Category of Containers

In Abbott et al. (2003), we began the investigation of functors which arise from containers by defining

$$T_{S \triangleright P} X \equiv \Sigma s : S.\, (P(s) \Rightarrow X)$$

where S is the object of shapes and P is the object of positions over S. We proved that such functors include constant functors, the identity and projections and are closed under fixed exponents, sums, products, least fixed points and greatest fixed points. These results are based upon two ideas:

1. To prove closure of container functors under least and greatest fixed points we generalised from containers and endofunctors (as described above) to containers in several parameters and functors $F : \mathbb{C}^n \to \mathbb{C}$.
2. We defined container morphisms and thereby obtain a category $\mathcal{G}$ of *containers*. T then becomes a functor $T : \mathcal{G} \to [\mathbb{C}, \mathbb{C}]$ which is full and faithful (when $[\mathbb{C}, \mathbb{C}]$ is understood as the category of *fibred* endo-functors and natural transformations) and which preserves limits, coproducts and filtered colimits of *cartesian diagrams*.

In the rest of this subsection we briefly describe these results so as to lay the groundwork for defining the derivatives of containers. The reader should consult Abbott et al. (2003) for further details and proofs.

To understand containers in several parameters, consider the bifunctor $1 + X \times Y$ and, in particular, how this bifunctor can be understood in terms of shapes and positions. Firstly, there are two shapes where data may be stored corresponding to the different summands of the coproduct. Above the first shape there are no positions whereas above the second shape there is one position for storing elements of X and one position for storing elements of Y. Crucially, where elements of X are stored is independent from where elements from Y are stored. This independence of how data is stored at one position from how data is stored at another position is a defining hallmark of containers. Consequently, a container in n variables in a locally cartesian closed category $\mathbb{C}$ is an object $A \in \mathbb{C}$ together with n objects over A, i.e. $A \vdash (B_i)_{i \in 1..n}$. The functor generated by such a container sends $\vec{X}$ to $\Sigma a : A . \prod_{i \in 1..n} (B_i(a) \Rightarrow X_i)$. For example, the bifunctor $1 + X \times Y$ arises with $n = 2$, $A = \{s_1, s_2\}$, $B_1(s_1) = B_2(s_1) = 0$ and $B_1(s_2) = B_2(s_2) = 1$.

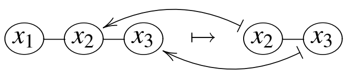
The second element of our analysis is the definition of container morphisms and the properties of the functor T from containers to functors.

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 $(\mathbb{C}/A)^I$.*

As an intuition for container morphisms, consider the 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) for each shape s in $\mathsf{List}\, X$ the choice of a shape $u(s)$ in $1 + \mathsf{List}\, X$; and ii) for each position above a shape $u(s)$, the choice of a position above s. This second component allows us to transform a labelling of the positions above a specific shape s to a labelling of the positions above $u(s)$. Notice the essential contravariance of this second component which results in a non-trivial mathematical theory. The first key result is that such container morphisms do indeed define natural transformations. More generally we can define:

Definition 3.2. *Define the functor $T : \mathcal{G}_I \to [\mathbb{C}^I, \mathbb{C}]$ taking each container $(A \triangleright B) \in \mathcal{G}_I$ to its "extension", the* container functor $T_{A \triangleright B}$, *as follows. For each $X \in \mathbb{C}^I$ define*

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

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:

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

A typical functor which is not in the image of T is $\neg\neg(X) = (X \to 0) \to 0$. The key properties of containers and the functor T were proved in Abbott et al. (2003) and are summarised below.

Theorem 3.3. *The following are true*

- *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 $\mathbb{C}$.*
- *The following containers define the identity, projection and constant functors*

  $$\mathrm{Id} \equiv (1 \triangleright 1) \quad \mathrm{Id}_j \equiv (1 \triangleright (\mathrm{Eq}(i,j))_{i \in I}) \quad K_C \equiv (C \triangleright 0)$$

  where $\mathrm{Eq}(i,j)$ *is* 1 *if $i = j$ and* 0 *otherwise.*

– *Containers are closed under sums, products and substitution as given by the formulae*

$$
\begin{aligned}
(A \rhd B) \times (C \rhd D) &\cong (A \times C \rhd \pi^* B + \pi'^* D) = (a : A,\, c : C \rhd B(a) + D(c)) \\
(A \rhd B) + (C \rhd D) &\cong (A + C \rhd B \mathbin{\mathring{+}} D) \\
(A \rhd B, E)[(C \rhd D)] &\cong \big(a : A, f : C^{E(a)} \rhd B(a) + \Sigma e : E(a). D(fe)\big)
\end{aligned}
$$

– *The functor $T : \mathscr{G}_I \to [\mathbb{C}^I, \mathbb{C}]$ is full and faithful and preserves limits and coproducts.*

Let us consider, for example, closure of containers under coproducts. Intuitively, if F and G are container functors generated by the containers $(A \rhd B)$ and $(C \rhd D)$, then an element of $F + G$ must be either an element from F or from G. Thus the shapes of $F + G$ must be $A + C$. In addition, the positions above a shape $\kappa(a)$ should just be $B(a)$ and similarly the positions above $\kappa'(c)$ should be $D(c)$. Thus the object of positions must be the global coproduct $B \mathbin{\mathring{+}} D$. A formal proof uses the fact that the formula defines the coproduct in $\mathscr{G}_I$ and that T preserves coproducts. A similar argument shows that containers are closed under products. This construction can be also found in Dybjer (1996) where it is used as an invariant in the proof that strictly positive types can be represented by W-types.

The substitution formula is used to construct chains (cochains) of containers which underpin their closure under least (greatest) fixed points. In particular, given a container in $\mathscr{G}_{I+1}$, we wish to fix the first I parameters and use substitution to build the relevant chain (cochain) whose colimit (limit) defines the least (greatest) fixed point. In effect, substitution is a functor $-[-] : \mathscr{G}_{I+1} \times \mathscr{G}_I \to \mathscr{G}_I$. When we substitute the $I+1$-parameter container $(A \rhd B, E)$ (where B is the I-indexed object of positions and E is the $I+1$'th object of positions) with the I-parameter container $(C \rhd D)$, we leave the B positions unaffected and replace each E-position with a shape in C above which data may be stored. Thus, the shapes of the resulting container will be a shape $a : A$ and a function $f : E(a) \Rightarrow C$. Above such a shape we will then have all the old B positions together with the D positions above fe for any $e : E(a)$.

Once we have substitution in place, we can defined fixed points by chains and cochains. Although $\mathscr{G}_I$ has colimits they are not in general preserved by T. This also applies to filtered colimits which underpin the construction of initial algebras. Fortunately, there exists a class of filtered colimits which is both sufficient for the construction of initial algebras and which *are* preserved by T. These diagrams are called *cartesian diagrams* as they are those diagrams which are built from *cartesian* morphisms defined as follows:

Definition 3.4. *A morphism $(u, f) : (A \rhd B) \to (C \rhd D)$ of containers is* cartesian *iff f is iso. The category of containers and cartesian morphisms is written $\widehat{\mathscr{G}}$.*

A natural transformation $\alpha : F \to G$ between two functors is cartesian *iff each naturality square of α is a pullback. The category of functors and cartesian natural transformations is written $\widehat{\mathscr{F}}$.*

In a cartesian morphism the map on positions merely rearranges them, and so morphisms in $\widehat{\mathscr{G}}$ cannot forget or copy data, hence they are in effect *linear* morphisms.

The following tells us that T restricts to a full and faithful functor $\widehat{T} : \widehat{\mathscr{G}} \to \widehat{\mathscr{F}}$.

Proposition 3.5. *A natural transformation $\alpha : T_F \to T_G$ between container functors is cartesian iff the corresponding container morphism (given by T full and faithful) is cartesian.* $\square$

Further, taking products restricts to a monoidal operation on $\widehat{\mathscr{G}}$ and $\widehat{\mathscr{F}}$ (as the projection $\pi : F \times G \to F$ is not cartesian, the monoidal operation is not the cartesian product).

Finally, we have all the elements in place to define least and greatest fixed points of containers. Given a container $(A \rhd B, E)$ in $\mathscr{G}_{I+1}$, substitution gives a functor $(A \rhd B, E)[-] : \mathscr{G}_I \to \mathscr{G}_I$ and hence we can define the least fixed point $\mu(A \rhd B, E)$ as the colimit of the chain

$$0 \longrightarrow (A \rhd B, E)[0] \longrightarrow (A \rhd B, E)^2[0] \longrightarrow \cdots$$

Crucially, this chain is a filtered cartesian diagram and hence its colimit is preserved by T. This has two important consequences i) our construction is correct:

$$T_{\bigvee (A \rhd B, E)^i[0]} \cong \bigvee T_{(A \rhd B, E)^i[0]}$$

and ii) the colimiting container $\bigvee (A \rhd B, E)^i[0]$ has a concrete description as

$$\bigvee (A \rhd B, E)^i[0] \cong \left(\bigvee A_i \rhd \bigvee B_i\right)$$

where A_i is the object of shapes of $(A \rhd B, E)^i[0]$ and B_i is the object of positions of $(A \rhd B, E)^i[0]$.

Greatest fixed points are easier. One uses substitution to define the appropriate cochain. Since $\mathscr{G}_I$ has all limits, $(A \rhd B, E)[-]$ preserves connected limits and T preserves all limits, containers are closed under greatest fixed points.

4 Derivatives of Containers

We now make the notion of derivative of a container explicit.

Definition 4.1. *If for containers F and H there exists a universal arrow in $\widehat{\mathscr{G}}$ from the functor $- \times H$ to F say that the* linear exponential of F by H *exists. Write the linear exponential as $H \multimap F$.*

In this situation we have a bijection of morphisms in $\widehat{\mathscr{G}}$

$$\frac{G \times H \longrightarrow F}{\overline{G \longrightarrow H \multimap F}} \ .$$

In this paper we investigate the special case when $H = \mathrm{Id}$.

Note that $\times$ here refers to the cartesian product in $\mathscr{G}$, it is just a tensor product in $\widehat{\mathscr{G}}$. Now, $\multimap$ gives us a convenient way to introduce the type of *one-hole contexts*:

Definition 4.2. *Say that a container F is* differentiable *iff the linear exponential* $\mathrm{Id} \multimap F$ *exists. Call this the* derivative *of F, written $\partial F \equiv \mathrm{Id} \multimap F$.*

Similarly for a container F in I arguments say that F is differentiable at $i \in I$ *iff* $\mathrm{Id}_i \multimap F$ *exists and write this as $\partial_i F$.*

The notion of *decidability* turns out to be central to the construction of derivatives.

Definition 4.3. *Say that an object $A \vdash B$ is* decidable *iff for each $a : A$ the equality relation on $B(a)$ is decidable, in other words $b, b' : B \vdash \mathrm{Eq}(b, b') + \neg\mathrm{Eq}(b, b') \cong 1$.*

Say that a container $F \equiv (A \triangleright B)$ is decidable *iff B is decidable. Say that a container $F \equiv (A \triangleright (B_i)_{i \in I})$ in I parameters is* decidable at j *iff B_j is decidable, and that F is* decidable *iff it is decidable at each $j \in I$.*

The following key theorem will be proved later on in section 4.2.

Theorem 4.4. *Every decidable container $F \equiv (A \triangleright B)$ is differentiable with derivative*

$$\partial(A \triangleright B) \cong (\Sigma_A B \triangleright B') \ .$$

Similarly, every container $F \equiv (A \triangleright (B_i)_{i \in I})$ decidable at j is differentiable with respect to j with derivative

$$\partial_j(A \triangleright (B_i)_{i \in I}) \cong (\Sigma_A B_j \triangleright (B_i'^j)_{i \in I}) \ ,$$

where $B_i'^j \equiv \pi^*_{B_j} B_i$ when $i \neq j$, and $B_i'^i \equiv B'$.

In the special case of a two parameter container $(A \triangleright B, E)$ this is

$$\partial_1(A \triangleright B, E) \cong (\Sigma_A B \triangleright B', \pi^*_B E) \qquad \partial_2(A \triangleright B, E) \cong (\Sigma_A E \triangleright \pi^*_E B, E') \ .$$

As we will show in section 4.2, the following facts about follow from theorem 4.4.

Proposition 4.5. *Derivatives of decidable containers satisfy the following*

$$\partial(F + G) \cong \partial F + \partial G \qquad\qquad \partial_i \mathrm{Id}_j \cong \mathrm{Eq}(i, j)$$
$$\partial(F \times G) \cong \partial F \times G + F \times \partial G \qquad\qquad \partial K \cong 0$$

and for F a container in 2 parameters

$$\partial(F[G]) \cong \partial_1 F[G] + \partial_2 F[G] \times \partial G$$
$$\partial \mu F \cong \mu(\partial_1 F[\mu F] + \partial_2 F[\mu F] \times \mathrm{Id}) \ .$$

We also conjecture that the derivative of the terminal coalgebra is given by

$$\partial \nu F \cong \mu(\partial_1 F[\nu F] + \partial_2 F[\nu F] \times \mathrm{Id}) \ .$$

Note that the derivative of a coalgebra is a *least* fixed point — this corresponds to the fact that a "hole" in the datatype must be accessible.

The use of containers in definition 4.2 is essential, we could not have derived the laws of calculus by using cartesian maps between functors instead. Indeed, if we assume that $\partial\mathrm{Id} = \mathrm{Id} \multimap \mathrm{Id} \cong 1$ for functors we can construct a cartesian morphism $\neg\neg \to 1$ which would imply (theorem 8.1, Abbott et al., 2003) that $\neg\neg$ is a container.

4.1 Properties of Decidable Objects

First we talk about decidable objects and their general properties. Decidability on B allows us to construct an object which acts as a kind of "complement" to elements of B.

Proposition 4.6. *If $A \vdash B$ is decidable then there exists an object $\Sigma_A B \vdash B'$, called the* complement of B, *satisfying the isomorphism $\pi_B^* B \cong 1 + B'$.*

Proof. Define $B' \equiv \Sigma_B \neg \mathrm{Eq}$ and calculate

$$\begin{aligned} A, b{:}B \vdash B &\cong \Sigma b'{:}B.\, 1 \cong \Sigma b'{:}B.\, (\mathrm{Eq}(b,b') + \neg\mathrm{Eq}(b,b')) \\ &\cong \Sigma b'{:}B.\, \mathrm{Eq}(b,b') + \Sigma b'{:}B.\, \neg\mathrm{Eq}(b,b') \cong 1 + B'(b) \ . \end{aligned}$$ □

The following lemma follows from the result that in an extensive category there is a cancellation rule $1 + A \cong 1 + B \implies A \cong B$.

Lemma 4.7. *If $A \vdash B$ is decidable and $\pi_B^* B \cong 1 + D$ then $B' \cong D$.* □

Some important properties of decidable objects follow.

Lemma 4.8. *If $A \vdash B$ is decidable then its complement $\Sigma_A B \vdash B'$ is decidable.*

Proof. Fixing $b{:}B$ then $a, c{:}B'(b)$ is equivalent to $a, c{:}B$ satisfying $a \neq b$ and $c \neq b$, and clearly decidability in B can then be used to compute decidability in B'. □

Lemma 4.9. *If $A \vdash B$ is decidable then for each $u{:}C \to A$ the pullback $u^* B$ is decidable with $(u^* B)' \cong u_B^* B'$.*

Proof. Observing that $b : u^* B$ can be written as $c : C \vdash b : B(uc)$ it is clear that decidability of B is inherited by $u^* B$. To verify the equation for $(u^* B)'$, calculate $\pi_{u^* B}^* u^* B \cong u_D^* \pi_B^* B \cong u_D^* (1 + B') \cong 1 + u_D^* B'$. □

Lemma 4.10. *Complements are closed under products and coproducts, and satisfy the following equations.*

$$\begin{aligned} &A + B \vdash (A+B)' \cong (A' + \pi_A^* B) \overset{\circ}{+} (\pi_B^* A + B') \\ &A \times B \vdash (A \times B)' \cong \pi^* A' + \pi'^* B' + \pi^* A' \times \pi'^* B' \\ &0 \vdash 0' \cong 0 \qquad 1 \vdash 1' \cong 0 \ . \end{aligned}$$

Proof. The results for $0'$ and $1'$ are immediate.

To show that $A + B$ is decidable, note first that since $\mathbb{C}$ is extensive we can analyse a binding $x{:}A + B$ into cases: $a{:}A \vdash x = \kappa a$ or $b{:}B \vdash y = \kappa' b$, and we know that $\kappa a \neq \kappa' b$ universally holds. Thus given $x, y{:}A + B$ decidability of equality reduces to decidability of equality separately in A and B.

To verify the complement of $A + B$ calculate:

$$\begin{aligned} \pi_{A+B}^*(A+B) &\cong \pi_{A+B}^* A + \pi_{A+B}^* B \cong \pi_A^*(A+B) \overset{\circ}{+} \pi_B^*(A+B) \\ &\cong (\pi_A^* A + \pi_A^* B) \overset{\circ}{+} (\pi_B^* A + \pi_B^* B) \\ &\cong (1 + A' + \pi_A^* B) \overset{\circ}{+} (\pi_B^* A + 1 + B') \\ &\cong 1 + ((A' + \pi_A^* B) \overset{\circ}{+} (\pi_B^* A + B')) \ . \end{aligned}$$

$A \times B$ is immediately decidable. The calculation of its complement is not needed in this paper and so is omitted. □

The corresponding results for Σ and $\overset{\circ}{+}$ require a little more attention to the base.

Lemma 4.11. *If $A \vdash B$ and $C \vdash D$ are both decidable then $A+C \vdash B \overset{\circ}{+} D$ is decidable with complement*

$$(\Sigma_{A+C}(B \overset{\circ}{+} D) \vdash (B \overset{\circ}{+} D)') \cong (\Sigma_A B + \Sigma_C D \vdash B' \overset{\circ}{+} D') \ .$$

If A is also decidable then so is $\Sigma_A B$ with complement

$$\Sigma_A B \vdash (\Sigma_A B)' \cong B' + \Sigma_{A'} B \ . \qquad \square$$

4.2 Theorems about Derivatives

We restate theorem 4.4 and prove it here by establishing the universal property.

Theorem 4.12. *If F is decidable (definition 4.3) then its derivative ∂F exists and is given by the formula*

$$\partial(A \triangleright B) \cong (\Sigma_A B \triangleright B') \ .$$

Proof. Let $F \equiv (A \triangleright B)$ be a decidable container and write $\phi : \pi_B^* B \cong 1 + B'$ for the isomorphism derived from the decidability of B. Write $dF \equiv (\Sigma_A B \triangleright B')$ and observe that $dF \times \mathrm{Id} \cong (\Sigma_A B \triangleright 1 + B')$. Now define the container morphism $@_F$ thus (here ϕ acts as a map of container positions, and is therefore contravariant to the shape map π_B):

$$@_F \equiv (\pi_B, \phi) : dF \times \mathrm{Id} \longrightarrow F \ ;$$

this map is obviously cartesian and is a map in $\widehat{\mathscr{G}}$. We will now show that $@_F$ is a universal arrow, and that therefore $dF \cong \partial F$ as required by the theorem.

Given a cartesian morphism $G \times \mathrm{Id} \to F$ where $G \equiv (C \triangleright D)$ by unrolling the definitions of morphisms in $\mathscr{G}$ this can understood as a pair of maps $(u : C \to A, f : 1 + D \cong u^* B)$, and the universal property of $@_F$ corresponds to a bijection

$$\frac{u : C \longrightarrow A \qquad f : 1 + D \cong u^* B}{v : C \longrightarrow \Sigma_A B \qquad g : D \cong v^* B'} \qquad \text{such that} \qquad \begin{array}{l} u = \pi_B \cdot v \\ f = v^* \phi \cdot (1 + g) \end{array} \ .$$

So given (u, f) for $c : C$ note that from the isomorphism $f_c : 1 + D(c) \cong B(uc)$ we can choose $b : B(uc)$ such that $D(c) \cong B(uc)'(b)$. This assignment $c \mapsto b$ together with the associated isomorphism gives us morphisms $v : C \to \Sigma_A C$ and $g : D \cong v^* B'$ as required, and it is clear from the construction that this assignment is unique. $\square$

We can now use the representation of derivatives established by this theorem to directly compute a number of important results about the derivatives of decidable containers.

To begin with, the following proposition tells us that if ∂F exists then so does $\partial^n F$ for every natural number n.

Proposition 4.13. *If a container is decidable then so is its derivative.*

Proof. This follows from lemma 4.8 that B' is decidable over B. $\square$

The elementary type operations interact with differentiation as one might expect.

Proposition 4.14. *Derivatives of decidable containers satisfy the following*

$$\partial(F+G) \cong \partial F + \partial G \qquad \partial(F \times G) \cong \partial F \times G + F \times \partial G$$
$$\partial K \cong 0 \qquad \partial_i \mathrm{Id}_j \cong \mathrm{Eq}(i,j)$$

The equations for $F+G$, $F \times G$ and K are unchanged if ∂ is replaced by ∂_i.

Proof. Each of these isomorphisms follows pretty directly from the results already proved for decidable objects. Let $F \equiv (A \rhd B)$, $G \equiv (C \rhd D)$, then:

$$\begin{aligned}\partial(F+G) &\cong \partial((A \rhd B) + (C \rhd D)) \cong \partial(A + C \rhd B \mathbin{♀} D) \\ &\cong (\Sigma_{A+C}(B \mathbin{♀} D) \rhd (B \mathbin{♀} D)') \cong (\Sigma_A B + \Sigma_C D \rhd B' \mathbin{♀} D') \\ &\cong (\Sigma_A B \rhd B') + (\Sigma_C D \rhd D') \cong \partial F + \partial G \ .\end{aligned}$$

The following derivation of $\partial(F \times G)$ omits both variables and explicit weakening, but these can be inferred unambiguously:

$$\begin{aligned}\partial(F \times G) &\cong \partial((A \rhd B) \times (C \rhd D)) \cong \partial(A \times C \rhd B + D) \\ &\cong (\Sigma_{A\times C}(B+D) \rhd (B+D)') \cong (\Sigma_{A\times C}(B+D) \rhd (B'+D) \mathbin{♀} (B+D')) \\ &\cong (\Sigma_{A\times C} B \rhd B' + D) + (\Sigma_{A\times C} D \rhd B + D') \\ &\cong (\Sigma_A B \rhd B') \times (C \rhd D) + (A \rhd B) \times (\Sigma_C D \rhd D') \cong \partial F \times G + F \times \partial G \ .\end{aligned}$$

A constant type $K \equiv (K \rhd 0)$ has derivative $\partial K = (\Sigma_K 0 \rhd 0') \cong (0 \rhd 0) \cong 0$. Similarly $\partial_i \mathrm{Id}_j = (\mathrm{Eq}(i,j) \rhd \mathrm{Eq}(i,j)') \cong (\mathrm{Eq}(i,j) \rhd 0) \cong \mathrm{Eq}(i,j)$. $\square$

An analogous form of the chain rule also holds for derivatives.

Proposition 4.15. *For decidable F and G the chain rule holds:*

$$\partial(F[G]) \cong \partial_1 F[G] + \partial_2 F[G] \times \partial G \ .$$

Proof. First note that in general

$$\begin{aligned}\partial(A \rhd B + C) &\cong (\Sigma_A(B+C) \rhd (B+C)') \\ &\cong (\Sigma_A(B+C) \rhd (B' + \pi_B^* C) \mathbin{♀} (\pi_C^* B + C')) \\ &\cong (\Sigma_A B \rhd B' + \pi_B^* C) + (\Sigma_A C \rhd \pi_C^* B + C') \ .\end{aligned}$$

Now let $F \equiv (A \rhd B, E)$ and $G \equiv (C \rhd D)$ and then (omitting explicit variables and weakening where possible)

$$\partial(F[G]) = \partial(\Sigma_A f : C^E \rhd B + \Sigma e : E.D(fe)) \cong (1) + (2)$$
$$\begin{aligned}\text{where}\quad (1) &\equiv (\Sigma_A \Sigma f : C^E.B \rhd B' + \Sigma e : E.D(fe)) \\ (2) &\equiv (\Sigma_A \Sigma f : C^E.\Sigma e : E.D(fe) \rhd B + (\Sigma e' : E.D(fe'))') \ .\end{aligned}$$

The first part, $(1) \cong \partial_1 F[G]$ follows pretty immediately:

$$\begin{aligned}(1) &\cong (\Sigma_A \Sigma_B f : C^E \rhd B' + \Sigma e : E.D(fe)) \\ &\cong (\Sigma_A B \rhd B', E)[(C \rhd D)] \cong \partial_1 F[G] \ .\end{aligned}$$

To reduce (2) to $\partial_2 F[G] \times \partial G$ it is necessary to be a little more explicit about the variables and context. First note in context A, $f : C^E$, $e : E$, $d : D(fe)$ that

$$(\Sigma e' : E . D(fe'))' \cong D(fe)'(d) + \Sigma e' : E' . D(fe') \ .$$

Next observe that we can rearrange the context to bring $e : E$ before $f : C^E$ and can then write $C^E \cong C^{E'+1} \cong C^{E'} \times C$ giving us new bindings $f' : C^{E'}$ and $c : C$. The function value fe must then be replaced by c, and so finally in context A, $e : E$, $f : C^{E'}$, $c : C$, $d : D(c)$ we can write

$$D(c)'(d) + \Sigma e' : E' . D(fe')$$

which is isomorphic (modulo the change of base just described) to $(\Sigma e' : E . D(fe'))'$. This now allows us to write

$$\begin{aligned}(2) &\cong (\Sigma_A \Sigma_E \Sigma f : C^{E'} . \Sigma_C D \triangleright B + D' + \Sigma e' : E' . D(fe')) \\ &\cong (\Sigma_A E \triangleright B, E')[(C \triangleright D)] \times (\Sigma_C D \triangleright D') \cong \partial_2 F[G] \times \partial G \ . \end{aligned}$$ □

The datatype μF arises as the iteration of the container substitution operation $F[-]$. If F has derivatives in both parameters (the fixed and the iterated parameter) then the derivative of μF can also be computed.

First we need the following technical lemma which will allow us to lift the construction of the μ type to the construction of derivatives.

Lemma 4.16. *The complement of a cartesian filtered colimit of decidable containers is the filtered colimit of their complements, hence* $\partial(\bigvee_i F_i) \cong \bigvee_i \partial F_i$.

Proof. Let each $F_i \equiv (A_i \triangleright B_i)$ and first observe that $\bigvee F \cong (\bigvee A \triangleright \bigvee B)$. Then for each $i \to j$ the following diagram can be constructed in $\mathbb{C}$:

$$\begin{tikzcd}
B'_i \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & B'_j \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \cdots \arrow[r] & \bigvee B' \arrow[d] \\
B_i \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & B_j \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \cdots \arrow[r] & \bigvee B \arrow[d] \\
A_i \arrow[r] & A_j \arrow[r] & \cdots \arrow[r] & \bigvee A
\end{tikzcd}$$

Since the bottom row consists of pullback squares then so does the top row (since the pullback of a complement is a complement), and thus $\bigvee \partial F$ is the top right hand arrow.

The object $\bigvee B'$ is the required complement of $\bigvee B$. □

Proposition 4.17. *If F is a decidable container in two parameters then the derivative of μF exists and is given by the equation*

$$\partial \mu F \cong \mu(\partial_1 F[\mu F] + \partial_2 F[\mu F] \times -) \ .$$

Proof. Define $G[-] \equiv \partial_1 F[\mu F] + \partial_2 F[\mu F] \times -$ and use the chain rule to observe

$$\partial \mu F \cong \partial(F[\mu F]) \cong \partial_1 F[\mu F] + \partial_2 F[\mu F] \times \partial \mu F = G[\partial \mu F] \ ;$$

this gives us a morphism $\theta : \mu G \to \partial \mu F$.

Conversely note that $\mu F \cong \bigvee_n (F^n[0])$ and so $\partial \mu F \cong \partial \bigvee_n (F^n[0]) \cong \bigvee_n \partial (F^n[0])$. We can therefore construct morphisms $\alpha_n : \partial(F^n[0]) \to G^n[0]$ inductively by setting

$$\alpha_{n+1} \equiv \partial_1 F[\kappa_n] + \partial_2 F[\kappa_n] \times \alpha_n$$

where $\kappa_n : F^n[0] \to \mu F$ is the colimiting cone of F. From the morphisms α we obtain a morphism $\phi : \partial \mu F \to \mu G$. It remains only to show that θ and ϕ are inverses. □

5 Abstract Containers

In future work we plan to include additional material on *abstract containers* and their derivatives, which we sketch below. Abstract containers arise when generalising the concept of a container such that includes structures where the order of elements does not matter, such as bags (or multisets). A bag is a list where sequences which can be obtained via an isomorphism on positions are identified. We follow the definition of analytic functors, as given in Hasegawa (2002), and allow any subgroup G of the automorphism group Aut_P of isomorphisms on P. Note however, that we work in a more general setting here as far as the ambient category is concerned and also that the type of positions is not necessarily finitely representable.

We introduce the notion of an *abstract container*[2] by the following data: an object of shapes S, a family $s : S \vdash P(s)$ and a family of subgroups of the automorphism groups $G(s) \subseteq \mathrm{Aut}_{P(s)}$ over $s : S$. We denote this abstract container by $(S \triangleright P/G)$. The associated functor $T_{S \triangleright P/G} : \mathbb{C} \to \mathbb{C}$ is defined by

$$T_{S \triangleright P/G} X \equiv \Sigma s : S. (P(s) \Rightarrow X / \sim_{G(s)}) \ .$$

where $f \sim_{G(s)} g \iff \exists \phi \in G(s). g = f \cdot \phi$. Categorically this can represented by a coequaliser. Morphisms between abstract containers $(A \triangleright B/G)$ and $(C \triangleright D/H)$ are given by an underlying container morphism $u : A \to C, a : A \vdash f : D(u(a)) \to B(a)$ such that $\forall \phi \in G(a). \exists \psi \in H(a). \phi \cdot f = f \cdot \psi$. We can extend T to a functor by noting that the extension of the underlying container morphism gives rise to a natural transformation between the abstract containers. We omit the straightforward generalisation to I-ary abstract containers. There is a full and faithful embedding from the category of containers to the category of abstract containers obtained by setting $G(s) = \{\mathrm{Id}_{P(s)}\}$. We believe that it is possible to extend all the properties shown for containers in Abbott et al. (2003) to abstract containers and additionally that abstract containers are closed under coequaliser of cartesian morphisms.

The notion of derivative as given before extends naturally to abstract containers, the derivative of an abstract container $F \equiv (A \triangleright B/G)$ is given by the derivative of the underlying container and by modifying G:

$$\partial F = \big(a : A, b : B(a) \triangleright \Sigma c : B(a). \neg \mathrm{Eq}(c, b) / G'(a, b)\big)$$

where $\phi \in G'(a,b) \iff \phi \in G(a) \wedge \phi(b) = b$. The characterisation of derivatives by $\partial F = \mathrm{Id} \multimap F$ extends to abstract containers.

[2] In analogy to abstract datatypes which are quotients of concrete datatypes.

The bag functor, as mentioned before, is represented by the abstract container

$$\mathrm{Bag} = (n : \mathbb{N} \triangleright n/\mathrm{Aut}_n)$$

Just by applying the rules of derivatives we obtain $\partial\mathrm{Bag} \cong \mathrm{Bag}$ and hence Bag plays the rule of e^x in calculus. This is maybe less surprising when noting that the extension of Bag

$$T_{\mathrm{Bag}}(X) = \Sigma n : \mathbb{N}. X^n/\mathrm{Aut}_n$$

corresponds to Euler's series defining e^x, since $|\mathrm{Aut}_n| = n!$.

Note, however, that one of the standard examples for abstract datatypes — finite sets — turns out not to be an abstract container. The reason for this is that the cardinality of positions depends on the equality of the argument type.

6 Conclusions

In the present paper, we have given an abstract characterisation of the derivative of a container, formalising the tangent of a container using a notion of linear function space. This should be useful in further understanding the relation between ordinary calculus and the calculus of containers developed here. We have shown that the usual laws of derivatives and McBride's law for least fixpoints can be derived from this notion using our previous work on containers (Abbott et al., 2003) and we have discussed abstract containers. We will make this precise and extend it to greatest fixpoints in further work.

References

M. Abbott, T. Altenkirch, and N. Ghani. Categories of containers. In *Proceedings of Foundations of Software Science and Computation Structures*, 2003.

J. Adámek and J. Rosický. *Locally Presentable and Accessible Categories*. Number 189 in LMS Lecture Note Series. CUP, 1994.

P. Dybjer. Representing inductively defined sets by wellorderings in Martin-Löf's type theory. *Theoretical Computer Science*, 170(1–2):245–276, 1996.

T. Ehrhard and L. Regnier. The differential lambda-calculus. URL http://iml.univ-mrs.fr/ftp/regnier/Articles.html. Available electronically, 2001.

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

M. Hofmann. On the interpretation of type theory in locally cartesian closed catetories. In *Computer Science Logic*, volume 933 of *LNCS*, 1994.

G. Huet. The zipper. *Journal of Functional Programming*, 7(5):549–554, 1997.

G. Huet. Linear contexts and the sharing functor: Techniques for symbolic computation. In *Workshop on Thirty Five Years of Automath*, 2003.

A. Joyal. Foncteurs analytiques et espéces de structures. In *Combinatoire énumérative*, number 1234 in LNM, pages 126 – 159. 1986.

G. Leibniz. Nova methodus pro maximis et minimis, itemque tangentibus, qua nec irrationals quantitates moratur. *Acta eruditorum*, 1684.

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