Decomposing Comonad Morphisms

Danel Ahman
Faculty of Physics and Mathematics, University of Ljubljana, Slovenia
danel.ahman@fmf.uni-lj.si

Tarmo Uustalu
Department of Computer Science, Reykjavik University, Iceland
Dept. of Software Science, Tallinn University of Technology, Estonia
tarmo@ru.is

Abstract

The analysis of set comonads whose underlying functor is a container functor in terms of directed containers makes it a simple observation that any morphism between two such comonads factors through a third one by two comonad morphisms, whereof the first is identity on shapes and the second is identity on positions in every shape. This observation turns out to generalize into a much more involved result about comonad morphisms to comonads whose underlying functor preserves Cartesian natural transformations to itself on any category with finite limits. The bijection between comonad coalgebras and comonad morphisms from costate comonads thus also yields a decomposition of comonad coalgebras.

2012 ACM Subject Classification Theory of computation → Logic; Theory of computation → Categorical semantics

Keywords and phrases container functors (polynomial functors), container comonads, comonad morphisms and comonad coalgebras, cofunctors, lenses

Digital Object Identifier 10.4230/LIPIcs.CALCO.2019.14

Funding *Danel Ahman*: This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-17-1-0326.
Tarmo Uustalu: Supported by the Icelandic Research Fund project grant no. 196323-051 and the Estonian Ministry of Education and Research institutional research grant no. IUT33-13.

Acknowledgements We thank our anonymous reviewers for very useful remarks.

1 Introduction

Containers of Abbott et al. [1] are a representation of a wide class of set functors (that one can use as parameterized datatypes) in terms of shapes and positions. Those set functors that enjoy this representation are called container functors. In joint work with Chapman [3], we found that container functors with a comonad structure can be characterized as interpretations of containers with corresponding additional structure, which we called directedness. In a directed container, every position in a shape determines another shape (its subshape), every shape has a designated root position, and positions in a subshape can be translated to the original shape. Remarkably, as we only noticed later [5], the category of directed containers is equivalent to the opposite of the category of small categories and cofunctors. Cofunctors were introduced by Aguiar [2]; they send objects from the target category to the source category, but maps from the source category to the target category.

Motivated by this equivalence, in this paper, we first show that an analogue of the full image factorization of functors holds for directed container morphisms: any directed container morphism decomposes into two whereby the first is identity on shapes and the second is identity on positions in every shape. Since the interpretation functor from the category of directed containers to the category of set comonads is fully-faithful, this immediately gives also a factorization of container comonad morphisms.

8th Conference on Algebra and Coalgebra in Computer Science (CALCO 2019).
Editors: Markus Roggenbach and Ana Sokolova; Article No. 14; pp. 14:1–14:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Then we ask if a similar decomposition is possible for general comonads on general categories. We show that only pullbacks and a terminal object (i.e., finite limits) are needed in order to formulate suitable substitutes for the notions of identity-on-shapes container morphism and identity-on-positions-in-every-shape container morphism for general natural transformations, and to obtain the factorization of general comonad morphisms.

That this decomposition is possible at this level of generality is nice, we find, since an alternative would have been to switch from containers to polynomials [7, 8]. At the basic level, containers and polynomials can be considered each other's notational variants, but the concepts of polynomials and polynomial functors scale to general categories with pullbacks [18]. However, already the definition of the polynomial analogue of the concept of directed container is complicated (we spelled it out in [3]), not to speak of the definition of the interpretation functor for it, or any proofs, so they are not the easiest to work with. The shapely types of Jay and Cockett [12, 11] are a lighter concept, but we did not need even those for our purpose.

The paper is organized as follows. In Section 2, we review containers and directed containers, including the equivalence of the category of directed containers to the opposite of the category of small categories and cofunctors. In Section 3, we describe our factorization of directed container morphisms or, which is the same, container comonad morphisms. In Section 4, we generalize this factorization to categories with finite limits. In Section 4 we also apply our results to the factorization of comonad coalgebras. We sum up in Section 5.

2 Preliminaries: containers and directed containers

We begin with a review of containers [1] and directed containers [3]. As noted above, containers are a representation for a certain class of set functors. Directed containers characterize, by additional structure on containers, those container functors that carry comonad structure.[1]

A *container* comprises a set S (of shapes) and, for any shape $s : S$, a set $P\,s$ (of positions in shape s). A *directed container* is a container (S, P) equipped with three maps

- $\downarrow : (\Sigma s : S.\, P\,s) \to S$ (the subshape corresponding to a position in a shape),
- $\mathsf{o} : \Pi_{s:S}.\, P\,s$ (the root position in a given shape), and
- $\oplus : \Pi_{s:S}.\, (\Sigma p : P\,s.\, P\,(s \downarrow p)) \to P\,s$ (translation of a position in a position's subshape)

satisfying the following five equations:

$$s \downarrow \mathsf{o}_s = s \qquad s \downarrow (p \oplus_s p') = (s \downarrow p) \downarrow p'$$

$$p \oplus_s \mathsf{o}_{s\downarrow p} = p \qquad \mathsf{o}_s \oplus_s p = p \qquad (p \oplus_s p') \oplus_s p'' = p \oplus_s (p' \oplus_{s\downarrow p} p'')$$

The 4th and 5th equations type because the 1st and 2nd hold. We note that the data and equations of a directed container are like those of a set, a monoid, and a right action of the monoid on the set, modulo the presence of the "minor" (subscripted) arguments and the dependent typing. In particular, if $P\,s$, o_s, and $p \oplus_s p'$ do not actually depend on s, then we indeed have a set, a monoid, and a right action of the monoid.

A container (S, P) defines a set functor $[\![S, P]\!]^{\mathrm{c}} = D$, called its *interpretation*, by

$$D\,X = \Sigma s : S.\, P\,s \Rightarrow X$$

[1] In what follows, subscript arguments of operations are "minor" arguments that can typically be inferred from the subsequent arguments. We generally write $\to$ for homsets, and $\Rightarrow$ for internal homs (exponential objects). In this and the next section, where we work in **Set**, this plays no role, but we still use the notation for conceptual clarity.

Given a directed container structure $(\downarrow, \mathsf{o}, \oplus)$ on (S, P), D obtains a comonad structure:

$$\varepsilon_X\,(s, v) = v\,\mathsf{o}_s \qquad \delta_X\,(s, v) = (s, \lambda p.\,(s \downarrow p, \lambda p'.\,v\,(p \oplus_s p')))$$

We call the comonad $[\![S, P, \downarrow, \mathsf{o}, \oplus]\!]^{\mathrm{dc}} = (D, \varepsilon, \delta)$ the *interpretation* of $(S, P, \downarrow, \mathsf{o}, \oplus)$.

Any comonad structure (ε, δ) on a set functor D that is the interpretation of some container (S, P) (i.e., $D\,X = \Sigma s : S.\,P\,s \Rightarrow X$) arises from a directed container structure $(\downarrow, \mathsf{o}, \oplus)$ on (S, P). In fact, directed container structures on (S, P) and comonad structures on D are in a bijection. Given a comonad structure (ε, δ), the corresponding directed container structure is defined by

$$\mathsf{o}_s = \varepsilon_{P\,s}\,(s, \mathsf{id}) \qquad s \downarrow p = \mathsf{fst}\,(\mathsf{snd}\,(\delta_{P\,s}\,(s, \mathsf{id}))\,p) \qquad p \oplus_s p' = \mathsf{snd}\,(\mathsf{snd}\,(\delta_{P\,s}\,(s, \mathsf{id}))\,p)\,p'$$

The following are some most prominent examples of directed containers with the corresponding comonads:

- Taking S to be any set, $P\,s = 1$, $s \downarrow * = s$, $\mathsf{o}_s = *$, $* \oplus_s * = *$, we get the *coreader comonad* defined by $DX = S \times X \cong \Sigma s : S.\,1 \Rightarrow X$, $\varepsilon\,(s, x) = x$, $\delta\,(s, x) = (s, (s, x))$.
- Taking S to be any set, $P\,s = S$, $s \downarrow s' = s'$, $\mathsf{o}_s = s$, $s' \oplus_s s'' = s''$, we get the *costate comonad* (also called the *array comonad* [16]) defined by $DX = S \times (S \Rightarrow X) \cong \Sigma s : S.\,S \Rightarrow X$, $\varepsilon\,(s, v) = v\,s$, $\delta\,(s, v) = (s, \lambda s'.(s', v))$.
- Choosing $S = 1$, $P\,* = \mathbb{N}$, $* \downarrow i = *$, $\mathsf{o}_* = 0$, $i \oplus_* j = i + j$, we obtain the *streams-with-suffixes comonad* defined by $DX = X^\omega \cong \Sigma s : 1.\,\mathbb{N} \Rightarrow X$, $\varepsilon\,(x_0, x_1, \ldots) = x_0$, $\delta\,(x_0, x_1, \ldots) = ((x_0, x_1, \ldots), (x_1, x_2, \ldots), \ldots)$.
- Choosing $S = \mathbb{N}$, $P\,n = [0..n]$, $n \downarrow i = s - i$, $\mathsf{o}_n = 0$, $i \oplus_n j = i + j$ gives us the *nonempty-lists-with-suffixes comonad* defined by $DX = X^+ \cong \Sigma n : \mathbb{N}.\,[0..n] \Rightarrow X$, $\varepsilon\,(x_0, x_1, \ldots, x_n) = x_0$, $\delta\,(x_0, x_1, \ldots, x_n) = ((x_0, x_1, \ldots, x_n), (x_1, x_2, \ldots, x_n), \ldots, (x_n))$.
- Take S to be the set of all bars where a *bar* (through the binary fan) is a finite set b of lists over $2 = \{0, 1\}$ such that any stream over 2 has exactly one prefix in b. Take $P\,b$ to be the set of all lists u over 2 that are a prefix of some list w in b. (A bar cuts a finite binary tree out of the infinite binary tree by establishing the positions of its leaves). Let $b \downarrow u = \{v \mid u \cdot v \in b\}$, $\mathsf{o}_b = ()$, $u \oplus_b v = u \cdot v$ (the empty list resp. concatenation of lists). This gives us the *labelled-finite-binary-trees comonad*. The counit extracts the label of the root node of the given tree. The comultiplication replaces the label of each node with the subtree rooted by that node.

Other useful examples of directed containers are obtained by constructions corresponding to the coproduct and product of two comonads, the cofree comonad on a functor, and compatible compositions of comonads (for all these, there are corresponding constructions of directed containers) and zipper datatypes (for those, there is a construction, called focussing, of turning any container (S, P) into a directed container whose shape set is $\Sigma s : S.\,P\,s$, i.e., its shapes are shapes of the given container together with a focus position) [3, 4].

A *morphism* between two *containers* (S, P) and (S', P') is given by maps $t : S \to S'$ (the shape map) and $q : \Pi_{s:S}.\,P'\,(t\,s) \to P\,s$ (the position map). A *morphism* between two *directed containers* $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$ is a morphism (t, q) between the underlying containers satisfying the following equations

$$t\,(s \downarrow q_s\,p) = t\,s \downarrow' p \qquad \mathsf{o}_s = q_s\,\mathsf{o}'_{t\,s} \qquad q_s\,p \oplus_s q_{s \downarrow q_s\,p}\,p' = q_s\,(p \oplus'_{t\,s} p')$$

Analogously to the interpretation of containers, a morphism (t, q) between containers (S, P) and (S', P') defines a natural transformation $[\![t, q]\!]^{\mathrm{c}} = \tau$ between their interpretations $[\![S, P]\!]^{\mathrm{c}} = D$ and $[\![S', P']\!]^{\mathrm{c}} = D'$ (the *interpretation* of (t, q)) by

$$\tau_X(s, v) = (t\,s, v \circ q_s)$$

Also, analogously to the interpretation of directed containers, if (t, q) is a morphism between directed containers $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$, then τ is a comonad morphism between $[\![S, P, \downarrow, \mathsf{o}, \oplus]\!]^{\mathrm{dc}} = (D, \varepsilon, \delta)$ and $[\![S', P', \downarrow', \mathsf{o}', \oplus']\!]^{\mathrm{dc}} = (D', \varepsilon', \delta')$; we define $[\![t, q]\!]^{\mathrm{dc}} = \tau$.

Any natural transformation τ between the interpretations D and D' of two containers (S, P) and (S', P') is an interpretation of a unique container morphism, namely (t, q) where

$$t\,s = \mathsf{fst}\,(\tau_{P\,s}(s, \mathsf{id})) \qquad q_s\,p = \mathsf{snd}\,(\tau_{P\,s}(s, \mathsf{id}))\,p$$

Furthermore, if τ is a comonad morphism between the interpretations (D, ε, δ) and $(D', \varepsilon', \delta')$ of two directed containers $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$, then (t, q) is a directed container morphism interpreting to τ.

Some examples of directed container morphisms are the following:

- Let $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$ be the directed containers for the costate comonad for S and coreader comonad for S, respectively. Take $t\,s = s$, $q_s * = s$. This corresponds to the comonad morphism $\tau_X : S \times (S \Rightarrow X) \to S \times X$ defined by $\tau\,(s, v) = (s, v\,s)$.
- Let $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$ be the directed containers for the nonempty lists comonad and the streams comonad, respectively. Take $t\,n = *$ and $q_n\,i = \min\,(i, n)$. This corresponds to the comonad morphism $\tau_X : X^+ \to X^\omega$ defined by $\tau\,(x_0, x_1, \ldots, x_n) = (x_0, x_1, \ldots, x_n, x_n, \ldots)$ (i.e., nonempty lists are padded out to streams).
- Let both $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$ be the directed container for the nonempty lists comonad. Let $t\,n = n \div 2$ and $q_n\,i = 2 * i$. This corresponds to the comonad morphism $\tau\,(x_0, x_1, \ldots, x_n) = (x_0, x_2, \ldots, x_{2*(n\div 2)})$ (i.e., every other element of a given nonempty list is dropped).
- Let $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$ be the directed containers for the nonempty lists comonad and the labelled finite binary trees comonad. Let $t\,n = \{w \in 2^* \mid |w| = n\}$ and $q_n\,u = |u|$. This corresponds to the comonad morphism sending a nonempty list xs to a labelled finite binary tree whose list of labels along any path is xs (so all paths have same length).
- Let $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$ be the directed containers for the labelled finite binary trees comonad and nonempty lists comonad. Let $t\,b$ be the length of the unique prefix in b of the stream 0^ω, i.e., the unique n such that $0^n \in b$. Let $q_b\,i = 0^i$. This directed container morphism (t, q) then represents the comonad morphism that maps a labelled finite binary tree to the non-empty list of labels along its leftmost path.

Containers and container morphisms form a monoidal category **Cont** (with a suitable container composition monoidal structure), and the interpretation of containers is a fully-faithful monoidal functor from **Cont** to $[\mathbf{Set}, \mathbf{Set}]$ (with the functor composition monoidal structure). Analogously, directed containers and directed container morphisms form a category **DCont**, and the interpretation of directed containers is fully-faithful functor from **DCont** to $\mathbf{Comonad}(\mathbf{Set})$. In fact, **DCont** is isomorphic to the category $\mathbf{Comonoid}(\mathbf{Cont})$ and is the pullback in **CAT** of $U : \mathbf{Comonad}(\mathbf{Set}) \to [\mathbf{Set}, \mathbf{Set}]$ along $[\![-]\!]^{\mathrm{c}} : \mathbf{Cont} \to [\mathbf{Set}, \mathbf{Set}]$.

In a sequel [5] to the first directed container work [3], we related directed containers to small categories. It turns out that directed containers are in a bijection up to isomorphism with small categories. Specifically, given a directed container $(S, P, \downarrow, \mathsf{o}, \oplus)$, the corresponding small category is obtained as follows. The set of objects is S. The set of maps with domain $s : S$ is $P\,s$, which means that the total set of maps is $\bar{P} = \Sigma s : S.\, P\,s$ and the domain of a map $(s, p) : \bar{P}$ is $\mathsf{src}\,p = s$. The codomain of a map $(s, p) : \bar{P}$ is $\mathsf{tgt}\,p = s \downarrow p$. The identity map on an object s is $\mathsf{id}_s = (s, \mathsf{o}_s)$ and the 1st directed container equation ensures that its codomain is $s \downarrow \mathsf{o}_s = s$, as required. A map (s, p) can only be composed with a map (s', p'), if $s \downarrow p = s'$, in which case the composition is $(s, p); (s', p') = (s, p \oplus_s p')$. By the 2nd directed container equation the codomain of this map is $s \downarrow (p \oplus_s p') = (s \downarrow p) \downarrow p'$, as required. The 3rd to the 5th equations then ensure that composition is unital and associative.

Of the above examples, the coreader comonad for S corresponds to the free category on a set of objects S, i.e., the discrete category (the only maps are the identity maps for every object). The costate comonad for S corresponds to the cofree category on a set of objects S, i.e., the codiscrete category (there is exactly one map between any two objects).

Although directed containers are in a bijection up to isomorphism with small categories, the category of directed containers is *not* equivalent to the category of small categories. The reason is that directed container morphisms are nothing like functors between small categories. Instead, they correspond to what Aguiar [2] has termed cofunctors, but with the source and target categories swapped.

A *cofunctor* between small categories $(S', \bar{P}', \mathsf{src}', \mathsf{tgt}', \mathsf{id}', ;')$ and $(S, \bar{P}, \mathsf{src}, \mathsf{tgt}, \mathsf{id}, ;)$ is given by two maps $t : S \to S'$ (the object map) and $\bar{q} : (\Sigma s : S.\, \Sigma p : \bar{P}'.\, t\,s = \mathsf{src}\,p) \to \bar{P}$ (the morphism map) satisfying $\mathsf{src}\,(\bar{q}\,(s, p)) = s$ and the following equations:

$$t\,(\mathsf{tgt}\,(\bar{q}\,(s,p))) = \mathsf{tgt}'\,p \qquad \mathsf{id}_s = \bar{q}\,(s, \mathsf{id}'_{t\,s}) \qquad \bar{q}\,(s,p)\,;\bar{q}\,(\mathsf{tgt}\,(\bar{q}\,(s,p)), p') = \bar{q}\,(s, p\,;'\,p')$$

While a functor maps objects and maps of the source category to those in the target category, a cofunctor's object map is from the target category to the source category, but the morphism map is still from the source to the target category.[2]

The category **DCont** of directed containers is equivalent to the opposite category of the category $\overleftarrow{\mathbf{Cat}}$ of small categories and cofunctors. Given a directed container morphism (t, q), the corresponding cofunctor is $(t, \bar{q})$ where $\bar{q}$ is defined by $\bar{q}\,(s, (t\,s, p)) = (s, q_s\,p)$.

Container functors with a monad structure can be also characterized in terms of additional structure on containers. This structure, studied by us [17] under the name of mnd-containers, is very different from directed containers. Mnd-containers can be seen as a version of nonsymmetric operads where operations may have infinite arities, arguments places of operations are identified nominally rather than positionally and arguments may be discarded and duplicated in composition.

3 Decomposing directed container morphisms

We now show that every morphism between two (directed) containers admits a natural factorization through a third (directed) container, an idea we promote to general functors and comonads in the next section.

It is almost immediate that every container morphism between two containers factorizes through a container with the shapes of the first and positions of the second container.

[2] A cofunctor looks a bit like a split opcleavage, but is not one. Before we learned about Aguiar's terminology, we spoke of a "relative split pre-opcleavage". See [6] for more discussion on this matter.

▶ **Proposition 1.** *Given two containers $C = (S, P)$, $C' = (S', P')$, a morphism $h = (t, q)$ between them factorizes through a third container C^* as below*

$$\begin{tikzcd} C \arrow[r, "h^1"'] \arrow[rr, bend left, "h"] & C^* \arrow[r, "h^2"'] & C' \end{tikzcd}$$

with the properties that

- $h^1 : C \to C^*$ *is identity on shapes and*
- $h^2 : C^* \to C'$ *is identity on positions in every shape.*

Proof. We define $C^* = (S^*, P^*)$ where $S^* = S$, $P^*\, s = P'\,(t\,s)$. I.e., C^* has the shapes of the first, but positions of the second container. The corresponding container morphisms are defined as $h^1 = (t^1, q^1)$ and $h^2 = (t^2, q^2)$ where $t^1\, s = s$, $q^1_s\, p = q_s\, p$, $t^2\, s = t\, s$, $q^2_s\, p = p$. ◀

At the level of functors and natural transformations, this is to say that a natural transformation $(\Sigma s : S.\, P\, s \Rightarrow -) \to (\Sigma s : S'.\, P'\, s \Rightarrow -)$, which we know must always be of the form $\lambda(s, v).\,(t\, s, v \circ q_s)$, always factors through the functor $\Sigma s : S.\, P'\,(t\, s) \Rightarrow -$.

Considerably more interestingly, this proposition can be strengthened to a factorization of any directed container morphism, in other words, of any morphism between two container comonads.

▶ **Proposition 2.** *If, in the situation of Proposition 1, C and C' come with directed container structures $(\downarrow, \mathsf{o}, \oplus)$ resp. $(\downarrow', \mathsf{o}', \oplus')$, and h is a directed container morphism, then C^* also carries a directed container structure, and h^1, h^2 are directed container morphisms.*

Proof. We define the directed container structure on C^* as $s \downarrow^* p = s \downarrow q_s p$, $\mathsf{o}^*_s = \mathsf{o}'_{t s}$, $p \oplus^*{}_s\, p' = p \oplus'_{t s}\, p'$. It is straightforward to verify that these data obey the laws a directed container:

$$s \downarrow^* \mathsf{o}^*_s = s \downarrow q_s \mathsf{o}'_{t s} = s \downarrow \mathsf{o}_s = s$$

$$\begin{aligned} s \downarrow^* (p \oplus^*{}_s\, p') &= s \downarrow q_s (p \oplus'_{t s}\, p') = s \downarrow (q_s p \oplus_s q_{s \downarrow q_s p} p') \\ &= (s \downarrow q_s p) \downarrow q_{s \downarrow q_s p} p' = (s \downarrow^* p) \downarrow q_{s \downarrow^* p} p' = (s \downarrow^* p) \downarrow^* p' \end{aligned}$$

$$\mathsf{o}^*_s \oplus^*{}_s\, p = \mathsf{o}'_{t s} \oplus'_{t s}\, p = p$$

$$p \oplus^*{}_s\, \mathsf{o}^*_{s \downarrow^* p} = p \oplus'_{t s}\, \mathsf{o}'_{t (s \downarrow^* p)} = p \oplus'_{t s}\, \mathsf{o}'_{t (s \downarrow q_s p)} = p \oplus'_{t s}\, \mathsf{o}'_{t s \downarrow' p} = p$$

$$\begin{aligned} (p \oplus^*{}_s\, p') \oplus^*{}_s\, p'' &= (p \oplus'_{t s}\, p') \oplus'_{t s}\, p'' = p \oplus'_{t s}\, (p' \oplus'_{t s \downarrow' p}\, p'') \\ &= p \oplus'_{t s}\, (p' \oplus'_{t (s \downarrow q_s p)}\, p'') = p \oplus^*{}_s\, (p' \oplus^*{}_{s \downarrow q_s p}\, p'') = p \oplus^*{}_s\, (p' \oplus^*{}_{s \downarrow^* p}\, p'') \end{aligned}$$

That h^1 and h^2 satisfy the directed container morphism laws is also straightforward. ◀

Let us now see what this means on our examples of directed container morphisms.

- Let $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$ be the directed containers for the nonempty lists comonad and the streams comonad respectively. Recall that we considered the directed container morphism given by $t\, n = *$ and $q_n\, i = \min\,(i, n)$. This directed container morphism factors through the directed container $(S^*, P^*, \downarrow^*, \mathsf{o}^*, \oplus^*)$ defined by $S^* = \mathbb{N}$, $P^*\, n = \mathbb{N}$, $n \downarrow^* i = n - \min\,(i, n)$, $\mathsf{o}^*_n = 0$, $i \oplus^*{}_n\, j = i + j$. This corresponds to the comonad defined by $D^*\, X = \mathbb{N} \times X^\omega \cong \Sigma n : \mathbb{N}.\, \mathbb{N} \Rightarrow X$, $\varepsilon^*\,(n, (x_0, x_1, \ldots)) = x_0$, $\delta^*\,(n, (x_0, x_1, \ldots)) = (n, ((n, (x_0, x_1, \ldots)), (n-1, (x_1, x_2, \ldots)), \ldots, (0, (x_n, x_{n+1}, \ldots)),$

$(0, (x_{n+1}, x_{n+2}, \ldots)), \ldots))$. It may be helpful to think of elements of this type as streams with a trusted initial segment: in the datastructure $(n, (x_0, x_1, \ldots))$, the elements $(x_0, x_1, \ldots, x_n)$ are trusted.

- Let both $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$ be the directed container for the nonempty lists comonad. Let us consider $t\,n = n \div 2$ and $q_n\,i = 2 * i$. This directed container morphism factors through the directed container $(S^*, P^*, \downarrow^*, \mathsf{o}^*, \oplus^*)$ defined by $S^* = \mathbb{N}$, $P^*\,n = [0..n \div 2]$, $n \downarrow^* i = n - 2 * i$, $\mathsf{o}^*_n = 0$, $i \oplus^*{}_n\, j = i + j$. This corresponds to the comonad defined by $D^* X = 2 \times X^+ \cong \Sigma n : \mathbb{N}.\, [0..n \div 2] \Rightarrow X$, $\varepsilon^*\,(b, (x_0, x_1, \ldots, x_m)) = x_0$, $\delta^*\,(b, (x_0, x_1, \ldots, x_m)) = (b, (b, (x_0, x_1, \ldots, x_m)), (b, (x_1, \ldots, x_m)), \ldots, (b, (x_m)))$. Here the thinking is that to recover n from $m = n \div 2$, one has to also know the parity b of n.
- Let $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$ be the directed containers for the nonempty lists comonad and the labelled finite binary trees comonad. Let $t\,n = \{w \in 2^* \mid |w| = n\}$ and $q_n\,u = |u|$. This directed container morphism factors through the directed container $(S^*, P^*, \downarrow^*, \mathsf{o}^*, \oplus^*)$ defined by $S^* = \mathbb{N}$, $P^*\,n = \{u \in 2^* \mid |u| \leq n\}$, $n \downarrow^* u = n - |u|$, $\mathsf{o}^*_n = ()$ and $u \oplus^*{}_n\, v = u \cdot v$. This is the comonad of labelled perfectly balanced binary trees.
- Let $(S, P, \downarrow, \mathsf{o}, \oplus)$ and $(S', P', \downarrow', \mathsf{o}', \oplus')$ be the directed containers for the labelled finite binary trees comonad and nonempty lists comonad. Let $t\,b$ be the length of the unique prefix in b of the stream 0^ω, i.e., the unique n such that $0^n \in b$. Let $q_b\,i = 0^i$. The directed container morphism (t, q) factors through the directed container $(S^*, P^*, \downarrow^*, \mathsf{o}^*, \oplus^*)$ defined by $S^* =$ "bars", $P^*\,b = [0..t\,b]$, $b \downarrow^* i = \{v \mid 0^i \cdot v \in b\}$, $\mathsf{o}^*_b = 0$, $i \oplus^*{}_b\, j = i + j$. This is a comonad of finite binary trees labelled along the leftmost path only. The counit extracts the label of the root node of the given tree. The comultiplication replaces the label of each node on the leftmost path with the subtree rooted by that node.

That the above-described factorization of container comonad morphisms should be possible is curious and by no means "granted". Strengthening the factorization of Proposition 1 to morphisms between container monads, for instance, does not work: the middle container functor is generally not a monad and the two natural transformations are not monad morphisms. Indeed, should C, C' be mnd-containers in the sense of [17], C^* will in general not be a mnd-container. For it to be one, from the given operations $\bullet : (\Sigma s : S.\, P\,s \Rightarrow S) \to S$ and $\bullet' : (\Sigma s : S'.\, P'\,s \Rightarrow S') \to S'$ (the shape maps for the multiplications of the corresponding monads), we would need to produce an operation $\bullet^* : (\Sigma s : S.\, P'\,(t\,s) \Rightarrow S) \to S$ (the shape map for the multiplication of a hypothetical middle monad). To define such an operation $\bullet^*$ in terms of $\bullet$, we would need a way to turn a given function $v : P'\,(t\,s) \to S$ into a function $v' : P\,s \to S$, but we cannot, since we cannot invert $q_s : P'\,(t\,s) \to P\,s$. To define $\bullet^*$ in terms of $\bullet'$ we would need to be able to convert a given shape $s : S'$ into a shape $s' : S$, but we cannot invert $t : S \to S'$ in general.

Let us also note that, in the light of the equivalence of **DCont** and $(\overleftarrow{\mathbf{Cat}})^{\mathrm{op}}$, our factorization of directed container morphisms is reminiscent of the full image factorization of functors [15]. In fact, it was the full image factorization that first lead us to the above factorization of directed container morphisms. Specifically, given a functor $F : \mathcal{C} \to \mathcal{D}$, its *full image* is the category $\overline{\mathsf{im}}\,F$ with as objects the objects of $\mathcal{C}$, and as morphisms $X \to Y$ the morphisms $FX \to FY$ of $\mathcal{D}$. The full image of F also comes with two functors: $\overline{F} : \mathcal{C} \to \overline{\mathsf{im}}\,F$ that acts as identity on objects and as F on morphisms, and $\overline{\overline{F}} : \overline{\mathsf{im}}\,F \to \mathcal{D}$ that acts as F on objects and as identity on morphisms. As such, $\overline{F}$ is the analogue of h^1 and $\overline{\overline{F}}$ the analogue of h^2 in the factorization of a directed container morphism h, as defined above. In the next section, we will see that we are indeed dealing with an analogue of full image factorization for cofunctors.

Given a container $C' = (S', P')$, a coalgebra structure with carrier S of $[\![C']\!]^{\mathrm{c}}$ is a map $\gamma : S \to \Sigma s : S'.\, P'\, s \Rightarrow S$, which splits into $t : S \to S'$ and $q : \Pi_{s:S}.\, P'\,(t\, s) \to S$. These are exactly the data of a container morphism from the costate container for S to C'. If $E' = (S', P', \downarrow', \mathsf{o}', \oplus')$ is a directed container, then γ is a coalgebra of the container comonad $[\![E']\!]^{\mathrm{dc}}$ iff t, q satisfy $t\,(q_s\, p) = t\, s \downarrow' p$, $s = q_s\, \mathsf{o}'_{t s}$, $q_{q_s p}\, p' = q_s\,(p \oplus'_{t s}\, p')$. These laws coincide with those of a directed container morphism from the costate directed container for S to E'.

Hence the factorization of morphisms between container functors (comonads) immediately gives us a factorization of container functor (comonad) coalgebra structures: a functor (comonad) coalgebra structure $\gamma : S \to \Sigma s : S'.\, P'\, s \Rightarrow S$ given by (t, q) factors as a composition of a functor (comonad) coalgebra structure $\gamma^* : S \to \Sigma s : S.\, P'\,(t\, s) \Rightarrow S$ given by (id_S, q) and a natural transformation (comonad morphism) given by $(t, \lambda_s.\, \mathsf{id}_{P'(t s)})$.

4 Decomposing general comonad morphisms

We now proceed to showing that the observations we made about morphisms between container comonads on **Set** hold about general comonads on general categories, under some assumptions. Specifically, they hold for comonad morphisms to comonads whose underlying functor preserves Cartesian natural transformations to itself on any category $\mathcal{C}$ with finite limits. For this, we first need to generalize identity-on-shapes and identity-on-positions-in-every-shape directed container morphisms to general comonad morphisms.

For an endofunctor D, which we think of as a datatype, we proceed from the idea that the shape of a datastructure in DX is its image under $D!_X$ in $D1$, which we treat as the object of shapes of D. A natural transformation $\phi : D \to D'$ between two datatypes can thus be considered bijective as a shape map if $\phi_1 : D1 \to D'1$ is an isomorphism.

We avoid introducing any objects of positions. We just think of a natural transformation $\psi : D^* \to D'$ as bijective as a position map for any shape in D^*1 and its image under ψ_1 in $D'1$ if ψ is Cartesian, i.e., if all its naturality squares

$$\begin{tikzcd} D^*X \arrow[r, "\psi_X"] \arrow[d, "D^*f"'] & D'X \arrow[d, "D'f"] \\ D^*Y \arrow[r, "\psi_Y"'] & D'Y \end{tikzcd}$$

are pullbacks. This is motivated by the following considerations. In the presence of a terminal object 1, it is sufficient (while trivially necessary) for Cartesianness of ψ that just the naturality squares for maps $!_X : X \to 1$, i.e.,

$$\begin{tikzcd} D^*X \arrow[r, "\psi_X"] \arrow[d, "D^*!_X"'] & D'X \arrow[d, "D'!_X"] \\ D^*1 \arrow[r, "\psi_1"'] & D'1 \end{tikzcd}$$

are pullbacks (cf. [14, Sec. 3.2]), because then, for any $f : X \to Y$, both the bottom square and outer square in the following diagram are pullbacks, and hence so is the top square, which is the naturality square for f:

$$\begin{tikzcd} D^*X \arrow[rr, "\psi_X"] \arrow[dr, "D^*f"] \arrow[ddr, bend right, "D^*!_X"'] & & D'X \arrow[dr, "D'f"] \arrow[ddr, bend right, "D'!_X"'] & \\ & D^*Y \arrow[rr, "\psi_Y"] \arrow[d, "D^*!_Y"] & & D'Y \arrow[d, "D'!_Y"] \\ & D^*1 \arrow[rr, "\psi_1"'] & & D'1 \end{tikzcd}$$

In the case of $\mathcal{C} = \mathbf{Set}$, the naturality square for $!_X$ being a pullback means that D^*X is isomorphic to the set of pairs $(s, xs) : D^*1 \times D'X$ such that $\psi_1\, s = D'\,!_X\, xs$, i.e., a shape s for D^* together with a datastructure xs in $D'X$ whose shape is the image of s under the shape map ψ_1. Since the map ψ_X is, up to this isomorphism, just the 2nd projection, and it is also natural in X, it must send datastructures in D^*X to datastructures in $D'X$ linearly, i.e., without discarding or duplicating any data (elements of X) contained in them.

We need to work with endofunctors preserving Cartesian natural transformations to themselves. We say that an endofunctor D' *preserves Cartesian natural transformations to* D' [3] if, for any endofunctor D and Cartesian natural transformation $\tau : D \to D'$, the natural transformation with components $D'\tau_X : D'DX \to D'D'X$ is also Cartesian. This may sound like a peculiar concept but was also needed by Kelly in his work on clubs and datatypes [14, Prop. 3.1]. Container functors have this property since they preserve arbitrary pullbacks.

We first show that natural transformations factorize as expected in the above sense.

▶ **Theorem 3** (cf. [14, Sec. 3.2])**.** *Given a category $\mathcal{C}$ with finite limits and two endofunctors D and D', a natural transformation τ from D to D' admits a factoring through a third endofunctor D^*, as depicted here,*

$$\begin{tikzcd}
DX \arrow[r, "\phi_X"'] \arrow[rr, bend left, "\tau_X"] & D^*X \arrow[r, "\psi_X"'] & D'X
\end{tikzcd}$$

with the properties that

- $\phi_1 : D1 \to D^*1$ *is an isomorphism and*
- $\psi : D^* \to D'$ *is Cartesian.*

Proof. For any X, we construct D^*X together with $\psi_X : D^*X \to D'X$ and $\pi_X : D^*X \to D1$ as a pullback. Further, we construct $\phi_X : DX \to D^*X$ as a unique map to this pullback.

$$\begin{tikzcd}
DX \arrow[dr, dotted, "\phi_X"] \arrow[drr, bend left, "\tau_X"] \arrow[ddr, bend right, "D!_X"'] \arrow[dr, phantom, shift right=4ex, "\boxed{\mathrm{C}}"] & & \\
& D^*X \arrow[r, "\psi_X"] \arrow[d, "\pi_X"'] \arrow[dr, phantom, "\boxed{\mathrm{A}}"] \arrow[ur, phantom, "\boxed{\mathrm{B}}"] & D'X \arrow[d, "D'!_X"] \\
& D1 \arrow[r, "\tau_1"'] & D'1
\end{tikzcd}$$

The latter construction presupposes commutation of the outer square above, which is immediate by the naturality of τ. Note that B gives us the desired factorization of τ.

For any $f : X \to Y$, we construct the map $D^*f : D^*X \to D^*Y$ as a unique map to the pullback D^*Y:

$$\begin{tikzcd}
D^*X \arrow[r, "\psi_X"] \arrow[dr, dotted, "D^*f"] \arrow[ddr, bend right, "\pi_X"'] \arrow[dr, phantom, shift right=4ex, "\boxed{\mathrm{E}}"] & D'X \arrow[dr, bend left, "D'f"] \arrow[d, phantom, "\boxed{\mathrm{D}}"] & \\
& D^*Y \arrow[r, "\psi_Y"] \arrow[d, "\pi_Y"'] \arrow[dr, phantom, "\mathrm{A}"] & D'Y \arrow[d, "D'!_Y"] \\
& D1 \arrow[r, "\tau_1"'] & D'1
\end{tikzcd}$$

[3] More precisely, composition with D' from the left preserves them. The terminology is from Garner [9].

This presupposes the commutativity of the outer square, which follows straightforwardly from A and uniqueness of maps to 1. We omit the identity and composition preservation proofs – these follow straightforwardly from id_{D^*X} and $D^*g \circ D^*f$ satisfying the same unique map properties as $D^*\,\mathsf{id}_X$ and $D^*(g \circ f)$.

The naturality square of ϕ for a map $f : X \to Y$ follows from both paths in it satisfying the properties of the unique map to the pullback D^*Y in the diagram

$$\begin{tikzcd}
DX \arrow[r, "Df"] \arrow[d, "\phi_X"'] \arrow[dr, dotted] & DY \arrow[d, "\phi_Y"'] \arrow[dr, bend left, "\tau_Y"] \arrow[r, phantom, "\mathrm{B}"] & {} \\
D^*X \arrow[r, "D^*f"] \arrow[dr, bend right, "\pi_X"'] \arrow[r, phantom, "\mathrm{E}", shift right=4ex] & D^*Y \arrow[r, "\psi_Y"] \arrow[d, "\pi_Y"] \arrow[dr, phantom, "\mathrm{A}"] & D'Y \arrow[d, "D'!_Y"] \\
{} & D1 \arrow[r, "\tau_1"'] & D'1
\end{tikzcd}$$

The commutativity of the outer square above follows from C, the naturality of τ, and uniqueness of maps to 1.

The naturality of ψ and π are just D and E.

To show that $\phi_1 : D1 \to D^*1$ is an isomorphism, we prove $\pi_1 : D^*1 \to D1$ to be its inverse. That the equation $\pi_1 \circ \phi_1 = \mathsf{id}_{D1}$ holds is proved as follows:

$$\begin{tikzcd}
D1 \arrow[dr, "\phi_1"] \arrow[ddr, bend right, "D!_1"'] \arrow[ddr, bend right=60, equal] & \\
{} \arrow[r, phantom, "\mathrm{C}"] & D^*1 \arrow[d, "\pi_1"] \\
{} & D1
\end{tikzcd}$$

The equation $\phi_1 \circ \pi_1 = \mathsf{id}_{D^*1}$ holds because both sides satisfy the properties of the unique map to the pullback D^*1 in the diagram

$$\begin{tikzcd}
D^*1 \arrow[r, "\pi_1"] \arrow[d, "\pi_1"'] \arrow[dr, dotted] & D1 \arrow[dr, bend left, "\tau_1"] & {} \\
D1 \arrow[dr, bend right, "D!_1"'] & D^*1 \arrow[r, "\psi_1"] \arrow[d, "\pi_1"'] \arrow[dr, phantom, "\mathrm{A}"] & D'1 \arrow[d, "D'!_1"] \\
{} & D1 \arrow[r, "\tau_1"'] & D'1
\end{tikzcd}$$

where the outer square commutes because $D!_1 = \mathsf{id}_{D1}$ and $D'!_1 = \mathsf{id}_{D'1}$. Indeed, $\phi_1 \circ \pi_1$ makes the two triangles above commute as follows:

$$\begin{tikzcd}
D^*1 \arrow[dr, "\pi_1"] & & & \\
{} & D1 \arrow[dr, "\phi_1"] \arrow[drr, bend left, "\tau_1"] \arrow[ddr, bend right, "D!_1"'] & \mathrm{B} & \\
{} & \mathrm{C} & D^*1 \arrow[r, "\psi_1"'] \arrow[d, "\pi_1"] & D'1 \\
{} & & D1 &
\end{tikzcd}$$

And so does id_{D^*1}:

$$\begin{tikzcd}
D^*1 \arrow[r, "\pi_1"] \arrow[d, "\pi_1"'] \arrow[dr, equal] & D1 \arrow[rrd, bend left, "\tau_1"] \arrow[dr, phantom, "\mathrm{A}"] & & \\
D1 \arrow[dr, bend right, "D!_1"'] \arrow[dr, equal, bend left] & D^*1 \arrow[r, "\psi_1"'] \arrow[d, "\pi_1"] & D'1 \arrow[r, bend left, "D'!_1"] \arrow[r, equal, bend right] & D'1 \\
& D1 & &
\end{tikzcd}$$

Finally, we must show that ψ is Cartesian, i.e., that the naturality squares

$$\begin{tikzcd}
D^*X \arrow[r, "\psi_X"] \arrow[d, "D^*!_X"'] & D'X \arrow[d, "D'!_X"] \\
D^*1 \arrow[r, "\psi_1"'] & D'1
\end{tikzcd}$$

for $!_X : X \to 1$ are pullbacks. This follows from D^*X being a pullback if we replace the node $D1$ by D^*1, which we know to be isomorphic:

$$\begin{tikzcd}
& D^*X \arrow[r, "\psi_X"] \arrow[d, "\pi_X"] \arrow[dl, bend right, "D^*!_X"'] \arrow[dr, phantom, "\mathrm{A}"] & D'X \arrow[d, "D'!_X"] \\
\arrow[r, phantom, "\mathrm{E}"] & D1 \arrow[r, "\tau_1"] \arrow[dl, bend left, "\phi_1"] \arrow[d, phantom, "\mathrm{B}"] & D'1 \\
D^*1 \arrow[ur, bend left, "\pi_1"] \arrow[ur, phantom, "\cong"] \arrow[urr, bend right, "\psi_1"'] & &
\end{tikzcd}$$

◀

Next we establish that not only do natural transformations factorize, but comonad morphisms do as well.

▶ **Theorem 4.** *If, in the situation of Theorem 3, D' preserves Cartesian natural transformations to D', both D and D' carry a comonad structure, and τ is a comonad morphism, then the constructed functor D^* also carries a comonad structure, and ϕ and ψ are comonad morphisms.*

Proof. We define the counit ε^* straightforwardly by

$$\varepsilon^*_X = D^*X \xrightarrow{\psi_X} D'X \xrightarrow{\varepsilon'_X} X$$

We construct the comultiplication δ^* as a unique map to D^*D^*X as a pullback obtained by pasting three pullbacks (of those, the right upper one is a pullback because D' preserves Cartesian natural transformations to D'):

$$\begin{tikzcd}
D^*X \arrow[rr, "\psi_X"] \arrow[d, "\pi_X"'] \arrow[drr, dotted, "\delta^*_X"] & & D'X \arrow[drr, "\delta'_X"] \arrow[d, phantom, "\boxed{\mathrm{F}}"] & & \\
D1 \arrow[dr, "\delta_1"'] & \boxed{\mathrm{G}} & D^*D^*X \arrow[r, "\psi_{D^*X}"] \arrow[dd, "D^*\pi_X"'] \arrow[ddr, phantom, "\psi\ \mathrm{Cart.}"] & D'D^*X \arrow[r, "D'\psi_X"] \arrow[d, "D'D^*!_X"] \arrow[dd, bend right, "D'\pi_X"'] \arrow[dr, phantom, "\psi\ \mathrm{Cart.}"] & D'D'X \arrow[d, "D'D'!_X"] \\
& DD1 \arrow[dr, "\phi_{D1}"'] & \arrow[r, phantom, "\mathrm{E}"] & D'D^*1 \arrow[r, "D'\psi_1"] \arrow[d, "D'\pi_1"] \arrow[dr, phantom, "\pi_1\mathrm{iso}"] & D'D'1 \arrow[d, equal] \\
& & D^*D1 \arrow[r, "\psi_{D1}"'] & D'D1 \arrow[r, "D'\tau_1"'] & D'D'1
\end{tikzcd}$$

This presupposes that the outer square above commutes, which is proved as follows:

$$
\begin{tikzcd}
D^*X \arrow[rr, "\psi_X"] \arrow[d, "\pi_X"'] \arrow[drr, phantom, "\mathrm{A}"] & & D'X \arrow[d, "D'!_X"] \arrow[rr, "\delta'_X"] & & D'D'X \arrow[dd, "D'D'!_X"] \\
D1 \arrow[rr, "\tau_1"] \arrow[dr, "\delta_1"'] & & D'1 \arrow[ddrr, "\delta'_1"] \arrow[ur, phantom, "\delta'\text{ nat.}"] & & \\
& DD1 \arrow[dr, "\phi_{D1}"'] \arrow[drr, bend left, "\tau_{D1}"] \arrow[ur, phantom, "\tau\text{ pres. comul.}"] \arrow[drr, phantom, "\mathrm{B}"] & & & \\
& & D^*D1 \arrow[r, "\psi_{D1}"'] & D'D1 \arrow[r, "D'\tau_1"'] & D'D'1
\end{tikzcd}
$$

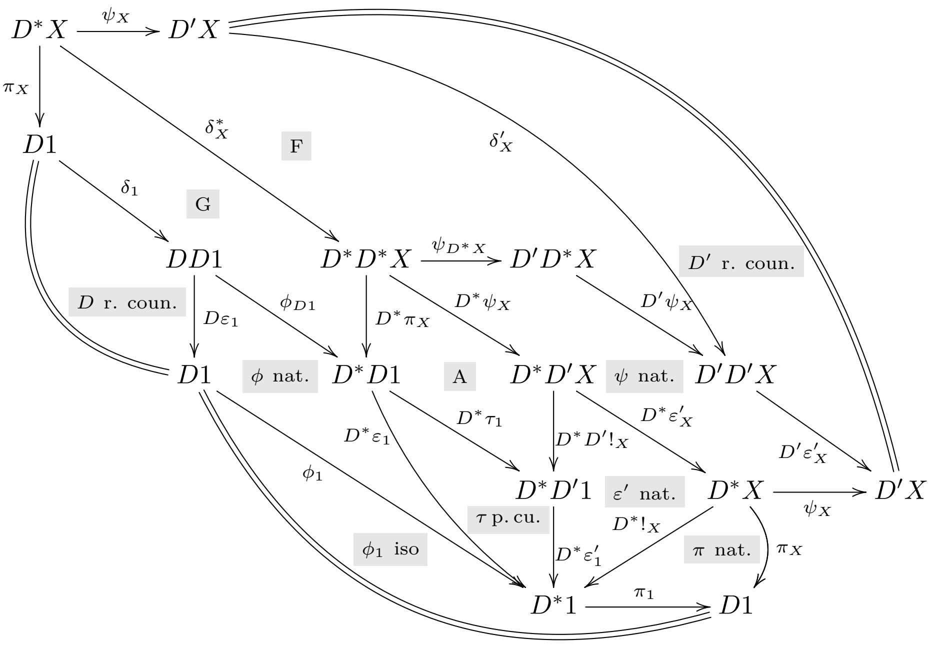
Next, we prove comonad laws for D^*. The counital laws $\varepsilon^*_{D^*X} \circ \delta^*_X = \mathsf{id}_{D^*X} = D^*\varepsilon^*_X \circ \delta^*_X$ hold because all three sides satisfy the properties of a unique map to the pullback D^*X:

$$
\begin{tikzcd}
D^*X \arrow[dr, dotted] \arrow[drr, bend left, "\psi_X"] \arrow[ddr, bend right, "\pi_X"'] & & \\
& D^*X \arrow[r, "\psi_X"] \arrow[d, "\pi_X"] \arrow[dr, phantom, "\mathrm{A}"] & D'X \arrow[d, "D'!_X"] \\
& D1 \arrow[r, "\tau_1"'] & D'1
\end{tikzcd}
$$

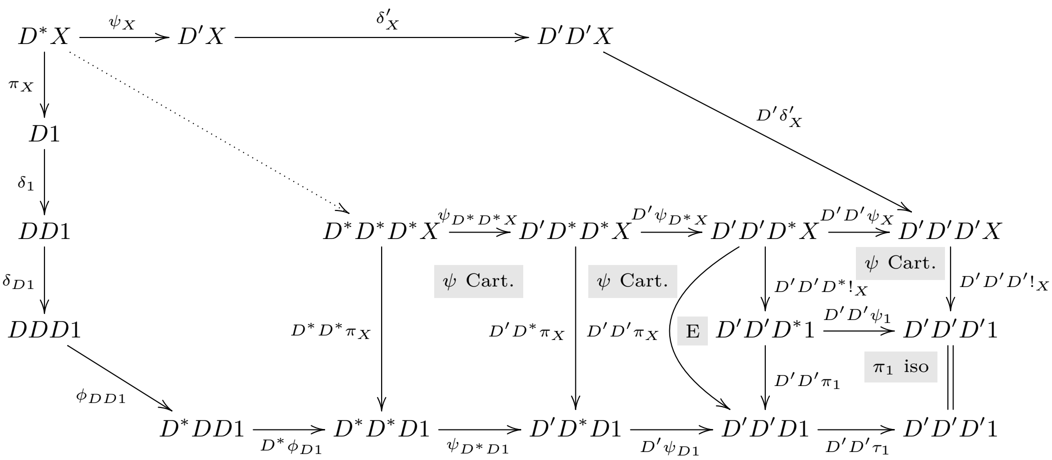
For id_{D^*X}, the two triangles above commute trivially. That they also commute for $\varepsilon^*_{D^*X} \circ \delta^*_X$ is proved as follows:

$$
\begin{tikzcd}
D^*X \arrow[r, "\psi_X"] \arrow[d, "\pi_X"'] \arrow[ddrr, "\delta^*_X"] & D'X \arrow[dddrrr, "\delta'_X"] \arrow[ddddrrrr, equals, bend left] \arrow[ddrr, phantom, "D'\text{ l. coun.}"] & & & & \\
D1 \arrow[dr, "\delta_1"] \arrow[ddddrrrr, equals, bend right] \arrow[r, phantom, "\mathrm{G}"] & & \arrow[r, phantom, "\mathrm{F}"] & & & \\
& DD1 \arrow[dr, "\phi_{D1}"] \arrow[ddrr, bend right, "\tau_{D1}"] \arrow[dddrrr, bend right, "\varepsilon_{D1}"'] \arrow[d, phantom, "D\text{ l. coun.}"] & D^*D^*X \arrow[d, "D^*\pi_X"] \arrow[dr, "\psi_{D^*X}"] & & & \\
& & D^*D1 \arrow[dr, "\psi_{D1}"] \arrow[r, phantom, "\psi\text{ nat.}"] \arrow[d, phantom, "\mathrm{B}"] & D'D^*X \arrow[r, "D'\psi_X"] \arrow[d, "D'\pi_X"] \arrow[dr, "\varepsilon'_{D^*X}"] & D'D'X \arrow[dr, "\varepsilon'_{D'X}"] \arrow[d, phantom, "\varepsilon'\text{ nat.}"] & \\
& & & D'D1 \arrow[dr, "\varepsilon'_{D1}"] \arrow[r, phantom, "\varepsilon'\text{ nat.}"] \arrow[d, phantom, "\tau\text{ pres. coun.}"] & D^*X \arrow[r, "\psi_X"'] \arrow[d, "\pi_X"] & D'X \\
& & & & D1 &
\end{tikzcd}
$$

And that they also commute for $D^*\varepsilon^*_X \circ \delta^*_X$ is proved as follows:



The coassociativity law $\delta^*_{D^*X} \circ \delta^*_X = D^*\delta^*_X \circ \delta^*_X$ holds because both sides satisfy the properties of a unique map to $D^*D^*D^*X$ as a pullback obtained by pasting together four pullbacks (of those, the middle and the right upper one are pullbacks since D' preserves Cartesian natural transformations to D'):



That $\delta^*_{D^*X} \circ \delta^*_X$ satisfies the two triangles in the above diagram is verified as follows:

$$
\begin{tikzcd}
D^*X \arrow[rrr, "\psi_X"] \arrow[d, "\pi_X"'] \arrow[dr, "\delta^*_X"] \arrow[drr, phantom, "\boxed{\mathrm{F}}"] & & & D'X \arrow[dr, "\delta'_X"] \arrow[drr, bend left, "\delta'_X"] \arrow[dr, phantom, "\boxed{D' \text{ coass.}}"'] & & \\
D1 \arrow[d, "\delta_1"'] \arrow[dr, phantom, "\boxed{\mathrm{G}}"] & D^*D^*X \arrow[rr, "\psi_{D^*X}"] \arrow[dr, "\delta^*_{D^*X}"] \arrow[dd, "D^*\pi_X"] & & D'D^*X \arrow[r, "D'\psi_X"] \arrow[dr, "\delta'_{D^*X}"] \arrow[dl, phantom, "\boxed{\mathrm{F}}"] & D'D'X \arrow[dr, "\delta'_{D'X}"] \arrow[d, phantom, "\boxed{\delta' \text{ nat.}}"] & D'D'X \arrow[d, "D'\delta'_X"] \\
DD1 \arrow[d, "\delta_{D1}"'] \arrow[dr, "\phi_{D1}"] & & D^*D^*D^*X \arrow[r, "\psi_{D^*D^*X}"'] \arrow[dd, "D^*D^*\pi_X"] \arrow[dl, phantom, "\boxed{\delta^* \text{ nat.}}"] & D'D^*D^*X \arrow[r, "D'\psi_{D^*X}"'] & D'D'D^*X \arrow[r, "D'D'\psi_X"'] & D'D'D'X \\
DDD1 \arrow[dr, bend right, "\phi_{DD1}"'] \arrow[r, phantom, "\boxed{\phi \text{ pres. comul.}}"] & D^*D1 \arrow[dr, "\delta^*_{D1}"] & & & & \\
& D^*DD1 \arrow[r, "D^*\phi_{D1}"'] & D^*D^*D1 & & &
\end{tikzcd}
$$

(This uses that ϕ preserves comultiplication, which is proved below.)

That also $D^*\delta^*_X \circ \delta^*_X$ satisfies the same diagrams is checked as follows:

$$
\begin{tikzcd}
& D^*X \arrow[rrrr, "\psi_X"] \arrow[d, "\pi_X"'] \arrow[dr, "\delta^*_X"] \arrow[drrr, phantom, "\boxed{\mathrm{F}}"] & & & & D'X \arrow[dr, "\delta'_X"] & \\
& D1 \arrow[dl, bend right, "\delta_1"'] \arrow[d, "\delta_1"] \arrow[dl, phantom, "\boxed{D \text{ coass.}}"] \arrow[dr, phantom, "\boxed{\mathrm{G}}"] & D^*D^*X \arrow[r, "\psi_{D^*X}"] \arrow[dr, "D^*\delta^*_X"'] \arrow[dd, "D^*\pi_X"] & D'D^*X \arrow[rrr, "D'\psi_X"] \arrow[dr, "D'\delta^*_X"] \arrow[d, phantom, "\boxed{\psi \text{ nat.}}"] & & \arrow[d, phantom, "\boxed{\mathrm{F}}"] & D'D'X \arrow[d, "D'\delta'X"] \\
DD1 \arrow[dr, bend right, "\delta_{D1}"'] & DD1 \arrow[d, "D\delta_1"'] \arrow[dr, "\phi_{D1}"] & & D^*D^*D^*X \arrow[r, "\psi_{D^*D^*X}"'] \arrow[dd, "D^*D^*\pi_X"] \arrow[dl, phantom, "\boxed{\mathrm{G}}"] & D'D^*D^*X \arrow[r, "D'\psi_{D^*X}"'] & D'D'D^*X \arrow[r, "D'D'\psi_X"'] & D'D'D'X \\
& DDD1 \arrow[dr, "\phi_{DD1}"'] \arrow[r, phantom, "\boxed{\phi \text{ nat.}}"] & D^*D1 \arrow[d, "D^*\delta_1"] & & & & \\
& & D^*DD1 \arrow[r, "D^*\phi_{D1}"'] & D^*D^*D1 & & &
\end{tikzcd}
$$

That ψ is a comonad morphism is straightforward. Indeed, the counit preservation law holds by the definition of ε^* while equation $\boxed{\mathrm{F}}$ is the comultiplication preservation law.

It remains to prove that ϕ is also a comonad morphism.

The counit preservation law $\varepsilon^* \circ \phi = \varepsilon$ is proved as follows:

$$
\begin{tikzcd}
DX \arrow[r, "\phi_X"] \arrow[rr, bend right, "\tau_X"] \arrow[dr, bend right=40, "\varepsilon_X"'] \arrow[r, phantom, bend right, "\boxed{\mathrm{B}}"'] & D^*X \arrow[r, "\psi_X"] \arrow[d, phantom, "\boxed{\tau \text{ pr. cu.}}"] & D'X \arrow[dl, bend left=40, "\varepsilon'_X"] \\
& X &
\end{tikzcd}
$$

The comultiplication preservation law $\delta^*_X \circ \phi_X = D^*\phi_X \circ \phi_{DX} \circ \delta_X$ holds because both the left-hand and right-hand sides satisfy the properties of a unique map to D^*D^*X:

$$
\begin{tikzcd}
DX \arrow[rr, "\tau_X"] \arrow[d, "D!_X"'] \arrow[drr, dotted] & & D'X \arrow[drr, "\delta'_X"] & & \\
D1 \arrow[dr, "\delta_1"'] & & D^*D^*X \arrow[r, "\psi_{D^*X}"] \arrow[dd, "D^*\pi_X"'] \arrow[r, phantom, shift right=4ex, "\psi\ \mathrm{Cart.}"] & D'D^*X \arrow[r, "D'\psi_X"] \arrow[d, "D'D^*!_X"] \arrow[dd, bend right, "D'\pi_X"'] \arrow[dr, phantom, "\psi\ \mathrm{Cart.}"] & D'D'X \arrow[d, "D'D'!_X"] \\
& DD1 \arrow[dr, "\phi_{D1}"'] & & D'D^*1 \arrow[r, "D'\psi_1"] \arrow[d, "D'\pi_1"] \arrow[dr, phantom, "\pi_1\mathrm{iso}"] \arrow[l, phantom, "\mathrm{E}"] & D'D'1 \arrow[d, equals] \\
& & D^*D1 \arrow[r, "\psi_{D1}"'] & D'D1 \arrow[r, "D'\tau_1"'] & D'D'1
\end{tikzcd}
$$

That $\delta^*_X \circ \phi_X$ satisfies the two triangles in the above diagram is verified as follows:

$$
\begin{tikzcd}
DX \arrow[dr, "\phi_X"] \arrow[rrrd, bend left, "\tau_X"] \arrow[ddr, bend right, "D!_X"'] & & & & \\
& D^*X \arrow[rr, "\psi_X"] \arrow[d, "\pi_X"] \arrow[dr, "\delta^*_X"] \arrow[l, phantom, "\mathrm{C}"] \arrow[ur, phantom, "\mathrm{B}"] & & D'X \arrow[dr, "\delta'_X"] \arrow[d, phantom, "\mathrm{F}"] & \\
& D1 \arrow[d, "\delta_1"'] \arrow[dr, phantom, "\mathrm{G}"] & D^*D^*X \arrow[r, "\psi_{D^*X}"'] \arrow[dd, "D^*\pi_X"] & D'D^*X \arrow[r, "D'\psi_X"'] & D'D'X \\
& DD1 \arrow[dr, "\phi_{D1}"'] & & & \\
& & D^*D1 & &
\end{tikzcd}
$$

That $D^*\phi_X \circ \phi_{DX} \circ \delta_X$ also satisfies the same two triangles is checked as follows:

$$
\begin{tikzcd}
DX \arrow[rr, "\tau_X"] \arrow[dr, "\delta_X"] \arrow[dd, "D!_X"'] & & D'X \arrow[dddrrr, bend left, "\delta'_X"] & & & \\
& DDX \arrow[dr, "\phi_{DX}"] \arrow[drr, bend left, "\tau_{DX}"] \arrow[dd, "DD!_X"] \arrow[l, phantom, "\delta\ \mathrm{nat.}"] \arrow[urr, phantom, "\tau\ \mathrm{pres.\ comul.}"] & & & & \\
D1 \arrow[dr, "\delta_1"'] & & D^*DX \arrow[r, "\psi_{DX}"] \arrow[dr, "D^*\phi_X"'] \arrow[ddr, bend right, "D^*D!_X"'] \arrow[u, phantom, "\mathrm{B}"] \arrow[d, phantom, "\phi\ \mathrm{nat.}"] & D'DX \arrow[dr, "D'\phi_X"] \arrow[drr, bend left, "D'\tau_X"] \arrow[d, phantom, "\psi\ \mathrm{nat.}"] & \mathrm{B} & \\
& DD1 \arrow[drr, "\phi_{D1}"'] & \mathrm{C} & D^*D^*X \arrow[r, "\psi_{D^*X}"'] \arrow[d, "D^*\pi_X"] & D'D^*X \arrow[r, "D'\psi_X"'] & D'D'X \\
& & & D^*D1 & &
\end{tikzcd}
$$

◀

Let us briefly compare the situation of Theorem 4 with the full image factorization of functors discussed in Section 3. Given a functor $F : \mathcal{C} \to \mathcal{D}$, the category $\overline{\mathsf{im}}\,F$, together with the associated functors $\overline{F}$ and $\overline{\overline{F}}$, arises as in the following pullback diagram in $\mathbf{Cat}$:

$$
\begin{tikzcd}
\mathcal{C} \arrow[dr, dotted, "\overline{F}"] \arrow[drr, bend left, "F"] \arrow[ddr, bend right, "!^{\mathcal{C}}"'] & & \\
& \overline{\mathsf{im}}\,F \arrow[r, "\overline{\overline{F}}"] \arrow[d] & \mathcal{D} \arrow[d, "!^{\mathcal{D}}"] \\
& \mathsf{codisc}(\mathcal{C}_0) \arrow[r, "\mathsf{codisc}(F_0)"'] & \mathsf{codisc}(\mathcal{D}_0)
\end{tikzcd}
$$

where $\mathsf{codisc}(\mathcal{C}_0)$ is the codiscrete category on the set of objects of $\mathcal{C}$ (the cofree category). The arrows $!^{\mathcal{C}}$ and $!^{\mathcal{D}}$ are the unique identity-on-objects functors.

We are dealing with the following pullback diagram in $\mathbf{Comonad}(\mathcal{C})$:

$$\begin{tikzcd}
D \arrow[dr, dotted, "\phi"] \arrow[drr, bend left, "\tau"] \arrow[ddr, bend right, "\langle D!,\varepsilon\rangle"'] & & \\
& D^* \arrow[r, "\psi"] \arrow[d] & D' \arrow[d, "\langle D'!,\varepsilon'\rangle"] \\
& D1\times - \arrow[r, "\tau_1\times -"'] & D'1\times -
\end{tikzcd}$$

This is obtained from the first diagram in the proof of Theorem 3, read as a diagram in $[\mathcal{C},\mathcal{C}]$ rather than $\mathcal{C}$, by replacing the constant functors $D1$ and $D'1$ by the corresponding cofree comonads (the coreader comonads for $D1$ and $D'1$). The special case for container comonads is, in the view of the equivalence of $\mathbf{DCont}$ and $(\overleftarrow{\mathbf{Cat}})^{\mathrm{op}}$, an analogue of full image factorization for cofunctors: a pushout diagram in $\overleftarrow{\mathbf{Cat}}$ involving discrete categories.

We do not prove it here, but the factorization asserted in Theorem 3 is unique up to a unique natural isomorphism (cf. [14, Sec. 3.2]). The factorization of Theorem 4 is unique up to a unique isomorphism of comonads. Thus in fact we have factorization systems on $[\mathcal{C},\mathcal{C}]$ and on the full subcategory of $\mathbf{Comonad}(\mathcal{C})$ given by underlying functors preserving Cartesian natural transformations to themselves. The "epis" of these factorization systems are natural transformations resp. comonad morphisms ϕ such that ϕ_1 is an isomorphism; the "monos" are Cartesian natural transformations resp. comonad morphisms.

We conclude by specializing the above results to the factorization of functor coalgebras and comonad coalgebras. This uses the costate functor and costate comonad.

▶ **Proposition 5.** *In a Cartesian closed category $\mathcal{C}$, given an object S, the functor $D^S = S \times (S \Rightarrow -)$ (the* costate functor *for S) carries a comonad structure (the* costate comonad*).*

Proof. Immediate from the fact that D^S is defined as the composition of the adjoint functors $S \times -$ and $S \Rightarrow -$. Accordingly, the counit and comultiplication ε^S and δ^S are constructed from the counit and unit of the adjunction: $\varepsilon^S_X = \mathsf{ev}_{S,X} : S \times (S \Rightarrow X) \to X$, $\delta^S_X = S \times \mathsf{coev}_{S,S\Rightarrow X} : S \times (S \Rightarrow X) \to S \times (S \Rightarrow (S \times (S \Rightarrow X)))$. ◀

Coalgebras of functors (resp. comonads) are the same as natural transformations (resp. comonad morphisms) from the costate functor (resp. comonad). This result is analogous to the well-known result about algebras of functors (resp. monads) and natural transformations (resp. monad morphisms) to the continuation functor (resp. monad) [13, 10].

▶ **Proposition 6.**

1. *In a Cartesian closed category $\mathcal{C}$, given a strong functor D', there is a bijection between maps from S to $D'S$ and natural transformations from D^S to D'.*
2. *If D' is a comonad, the same bijection restricts to a bijection between comonad coalgebras of D' with carrier S and comonad morphisms from D^S to D'.*

Proof (sketch). We use that tensorially strong functors are internally functorial. We construct the bijection as follows.

Given a map $\gamma : S \to D'S$, we define a natural transformation $\tau : D^S \to D'$ by

$$\tau_X = \; S \times (S \Rightarrow X) \xrightarrow{\gamma\times\mathsf{ifunc}^{D'}_{S,X}} D'S \times (D'S \Rightarrow D'X) \xrightarrow{\mathsf{ev}_{D'S,D'X}} D'X$$

If D' is a comonad and γ satisfies the laws of a comonad coalgebra structure, then τ satisfies the laws of a comonad morphism.

Given a natural transformation $\tau : D^S \to D'$, we define a map $\gamma : S \to D'S$ by

$$\gamma = \; S \xrightarrow{\langle \mathsf{id}_S, !_S \rangle} S \times 1 \xrightarrow{S \times \mathsf{iid}_S} S \times (S \Rightarrow S) \xrightarrow{\tau_S} D'S$$

If D' is a comonad and τ satisfies the laws of a comonad morphism, then γ satisfies the laws of a comonad coalgebra structure.

The two transformations are mutual inverses. ◀

Using what we have learned about the costate functor and costate comonad, we obtain a decomposition of functor coalgebras and comonad coalgebras.

▶ **Theorem 7.**

1. *Given a Cartesian closed finitely complete category $\mathcal{C}$, a strong functor D' preserving Cartesian natural transformations to D', and a map $\gamma : S \to D'S$, then γ admits a factoring through the object D^*S for another functor D^*, as depicted below*

$$\begin{tikzcd} S \arrow[r, "\gamma^*"'] \arrow[rr, bend left, "\gamma"] & D^*S \arrow[r, "\psi_S"'] & D'S \end{tikzcd}$$

with the properties that
- $D^*! \circ \gamma^* : S \to D^*1$ *is an isomorphism and*
- $\psi : D^* \to D'$ *is Cartesian.*

2. *If D' is a comonad and γ is a comonad coalgebra structure, then D^* is a comonad, γ^* is a comonad coalgebra structure and ψ is a comonad morphism.*

Proof (sketch). This is a corollary of Theorems 3, 4 and the last two propositions.

The given map $\gamma : S \to D'S$ induces a natural transformation $\tau : D^S \to D'$. From this, we get a functor D^* and two natural transformations $\phi : D^S \to D^*$ and $\psi : D^* \to D'$, whereby $\phi_1 : D^S 1 \to D^*1$ is an isomorphism and ψ is Cartesian. We construct $\gamma^* : S \to D^*S$ as the composition $\phi_S \circ (S \times \mathsf{iid}_S) \circ \langle \mathsf{id}_S, !_S \rangle$. The map $D^*! \circ \gamma^*$ is an isomorphism thanks to commutation of the diagram

$$\begin{tikzcd} S \arrow[r, "\langle \mathsf{id}_S, !_S \rangle", "\cong"'] \arrow[rrr, bend left, "\gamma^*"] & S \times 1 \arrow[r, "S \times \mathsf{iid}_S"] \arrow[dr, "\cong"'] & S \times (S \Rightarrow S) \arrow[r, "\phi_S"] \arrow[d, "D^S!"] & D^*S \arrow[d, "D^*!"] \\ & & S \times (S \Rightarrow 1) \arrow[r, "\phi_1", "\cong"'] & D^*1 \end{tikzcd}$$

◀

5 Conclusion

We have demonstrated that two observations about comonads that are immediate for container comonads on **Set** also hold more generally for comonads whose underlying functor preserves Cartesian natural transformations to itself on any finitely complete category. These observations concern shapes and positions (in terms of comonad morphisms being bijective on shapes or bijective on positions between corresponding pairs of shapes), and demonstrate that comonads generally, not just container comonads, are usefully analyzed in terms of shapes and positions and exhibit noteworthy properties expressible in these terms.

In other work [6], we have shown that container comonad coalgebras and container comonad morphisms can be seen as generalized asymmetric (i.e., server-client) lenses, which are a device for keeping a client's view of a database in synch with the master copy at a server. Shapes in the two directed containers are states of the two databases, positions are updates. The factorization results presented in this paper say that such lenses factorize into two lenses, whereof the first is identity on states and the second is identity on updates for every state.

References

1 Michael Abbott, Thorsten Altenkirch, and Neil Ghani. Containers: Constructing Strictly Positive Types. *Theor. Comput. Sci.*, 342(1):3–27, 2005. doi:10.1016/j.tcs.2005.06.002.

2 Marcelo Aguiar. *Internal Categories and Quantum Groups.* PhD thesis, Cornell University, 1997. URL: http://www.math.cornell.edu/~maguiar/thesis2.pdf.

3 Danel Ahman, James Chapman, and Tarmo Uustalu. When Is a Container a Comonad? *Log. Methods Comput. Sci.*, 10(3), 2014. article 14. doi:10.2168/lmcs-10(3:14)2014.

4 Danel Ahman and Tarmo Uustalu. Distributive Laws of Directed Containers. *Progress in Informatics*, 10:3–18, 2013. doi:10.2201/niipi.2013.10.2.

5 Danel Ahman and Tarmo Uustalu. Directed Containers as Categories. In Robert Atkey and Neil Krishnaswami, editors, *Proc. of 6th Wksh. on Mathematically Structured Functional Programming, MSFP 2016 (Eindhoven, Apr. 2016)*, volume 207 of *Electron. Proc. in Theor. Comput. Sci.*, pages 89–98. Open Publishing Assoc., 2016. doi:10.4204/eptcs.207.5.

6 Danel Ahman and Tarmo Uustalu. Taking Updates Seriously. In Romina Eramo and Michael Johnson, editors, *Proc. of 6th Wksh. on Mathematically Structured Functional Programming, MSFP 2016 (Eindhoven, Apr. 2016)*, volume 1827 of *CEUR Wksh. Proc.*, pages 59–73. RWTH Aachen, 2017. URL: http://ceur-ws.org/Vol-1827/paper11.pdf.

7 Nicola Gambino and Martin Hyland. Wellfounded Trees and Dependent Polynomial Functors. In Stefano Berardi, Mario Coppo, and Ferruccio Damiani, editors, *Revised Selected Papers from Int. Wksh. on Types for Proofs and Programs, TYPES 2003 (Torino, Apr./May 2003)*, volume 3085 of *Lect. Notes in Comput. Sci.*, pages 210–225. Springer, 2004. doi:10.1007/978-3-540-24849-1_14.

8 Nicola Gambino and Joachim Kock. Polynomial Functors and Polynomial Monads. *Math. Proc. Cambridge Philos. Soc.*, 154(1):153–192, 2013. doi:10.1017/s0305004112000394.

9 Richard Garner. Double Clubs. *Cahiers de Topologie et Géométrie Différentielle Catégoriques,*, 47(4):261–316, 2006. URL: http://www.numdam.org/item?id=CTGDC_2006__47_4_261_0.

10 Martin Hyland, Paul B. Levy, Gordon Plotkin, and John Power. Combining Algebraic Effects with Continuations. *Theor. Comput. Sci.*, 375(1–3):20–40, 2007. doi:10.1016/j.tcs.2006.12.026.

11 C. Barry Jay. A Semantics for Shape. *Sci. Comput. Program.*, 25(2–3):251–283, 1995. doi:10.1016/0167-6423(95)00015-1.

12 C. Barry Jay and J. Robin B. Cockett. Shapely Types and Shape Polymorphism. In Donald Sannella, editor, *Proc. of 5th European Symp. on Programming Languages and Systems, ESOP '94 (Edinburgh, Apr. 1994)*, volume 788 of *Lect. Notes in Comput. Sci.*, pages 302–316. Springer, 2004. doi:10.1007/3-540-57880-3_20.

13 G. Max Kelly. A Unified Treatment of Transfinite Constructions for Free Algebras, Free monoids, Colimits, Associated Sheaves, and So on. *Bull. Austral. Math. Soc.*, 22(1):1–83, 1980. doi:10.1017/s0004972700006353.

14 G. Max Kelly. On Clubs and Data-Type Constructors. In Michael P. Fourman, Peter T. Johnstone, and Andrew M. Pitts, editors, *Applications of Categories in Computer Science*, volume 177 of *London. Math. Soc. Lect. Note Series*, pages 163–190. Cambridge Univ. Press, 1992. doi:10.1017/cbo9780511525902.010.

15 nLab authors. Full Image. nLab entry, revision 5, April 2016. URL: `http://ncatlab.org/nlab/revision/full%20image/5`.

16 John Power and Olha Shkaravska. From Comodels to Coalgebras: State and Arrays. *Electron. Notes Theor. Comput. Sci.*, 106:297–314, 2004. `doi:10.1016/j.entcs.2004.02.041`.

17 Tarmo Uustalu. Container Combinatorics: Monads and Lax Monoidal Functors. In Mohammad Reza Mousavi and Jiří Sgall, editors, *Proc. of 2nd IFIP WG 1.8 Int. Conf. on Topics in Theoretical Computer Science, TTCS 2017 (Tehran, Sept. 2017)*, volume 10608 of *Lect. Notes in Comput. Sci.*, pages 91–105. Springer, 2017. `doi:10.1007/978-3-319-68953-1_8`.

18 Mark Weber. Polynomials in Categories with Pullbacks. *Theor. Appl. Categ.*, 30(16):533–598, 2015. URL: `http://www.tac.mta.ca/tac/volumes/30/16/30-16abs.html`.