Theory and Applications of Categories, Vol. 30, No. 18, 2015, pp. 620–686.

CARTESIAN DIFFERENTIAL STORAGE CATEGORIES

R. BLUTE, J.R.B. COCKETT, AND R.A.G. SEELY

Abstract. Cartesian differential categories were introduced to provide an abstract axiomatization of categories of differentiable functions. The fundamental example is the category whose objects are Euclidean spaces and whose arrows are smooth maps.

Monoidal differential categories provide the framework for categorical models of differential linear logic. The coKleisli category of any monoidal differential category is always a Cartesian differential category. Cartesian differential categories, besides arising in this manner as coKleisli categories, occur in many different and quite independent ways. Thus, it was not obvious how to pass from Cartesian differential categories back to monoidal differential categories.

This paper provides natural conditions under which the linear maps of a Cartesian differential category form a monoidal differential category. This is a question of some practical importance as much of the machinery of modern differential geometry is based on models which implicitly allow such a passage, and thus the results and tools of the area tend to freely assume access to this structure.

The purpose of this paper is to make precise the connection between the two types of differential categories. As a prelude to this, however, it is convenient to have available a general theory which relates the behaviour of "linear" maps in Cartesian categories to the structure of Seely categories. The latter were developed to provide the categorical semantics for (fragments of) linear logic which use a "storage" modality. The general theory of storage, which underlies the results mentioned above, is developed in the opening sections of the paper and is then applied to the case of differential categories.

1. Introduction

A fundamental observation of Girard, [G87], which led to the development of linear logic, was that the hom-functor of stable domains could be decomposed as $A \Rightarrow B := \ !A \multimap B$ where the latter, $X \multimap Y$, is the hom-functor of coherence spaces. This suggested that one might similarly be able to decompose differentiable maps between two spaces A and B as the "linear" maps from a constructed space $S(A)$ to B. Ehrhard and Regnier's work [ER04] on the differential λ-calculus was inspired by this analogy and in [E04] Ehrhard provided concrete examples which realized this analogy. This raised the question of how general a correspondence this was.

Research partially supported by NSERC and the Centre International de Recontres Mathématiques, and by FRQNT.

Received by the editors 2014-09-29 and, in revised form, 2015-05-05.

Transmitted by Susan Niefield. Published on 2015-05-06.

2010 Mathematics Subject Classification: 18D10,18C20,12H05,32W99.

Key words and phrases: monoidal categories, differential categories, Kleisli categories, differential operators.

© R. Blute, J.R.B. Cockett, and R.A.G. Seely, 2015. Permission to copy for private use granted.

In [BCS06] the notion of a (monoidal) differential category was introduced as a minimal categorical doctrine in which differentiability could be studied. This generalized Ehrhard and Regnier's ideas in various ways. It dispensed with the necessity that the setting be $*$-autonomous: the decomposition of the hom-functor was then handled through the presence of a comonad—or modality—satisfying a minimal set of coherences. Finally the differential was introduced satisfying a minimal set of identities. In this paper we shall, in fact, consider a slight strengthening of this basic notion by requiring, first, that the modality be a storage modality (*i.e.* the Seely isomorphisms $1 \equiv S(0)$ and $S(A) \otimes S(B) \equiv S(A \times B)$ are present) and, second, that the differential satisfies an interchange law.

This maintained the perspective of linear logic, by providing a decomposition of differentiable functions through a comonad—or storage modality—on more basic "linear" functions. However, a rather unsatisfactory aspect of this perspective remained. Smooth functions (*i.e.* infinitely differentiable functions)—after all the main subject of calculus—appeared only indirectly as the maps of the coKleisli category. This left a veil between these settings and the direct understanding of these differentiable functions in the classical sense. It seemed, therefore, important to develop a more direct view of what a category of differentiable functions should look like. This caused us to examine more closely the coKleisli categories of differential categories and to seek a direct axiomatization for them.

To advance this aim we deployed (the dual of) Fuhrmann's notion of an abstract Kleisli category [F99]. In the course of developing these ideas we stumbled on a much deeper sense in which the situation in linear logic can be read as mathematics. Thus a key result of this paper is a very general structural theorem about how "models of linear logic" arise. Rather than retrospectively remove our path to this result, we have centred the paper around the results on differential categories so that the motivation which brought us to this particular passage is not lost.

The "models of linear logic" we consider here are not, in fact, models of full linear logic, as the underlying categories are monoidal and not, in general, monoidal closed, let alone $*$-autonomous. These models were studied in depth by Gavin Bierman [B95]: he called them "Seely categories". Given that these categories have now been the object of quite a number of studies, it is perhaps surprising that there is more to say about them. Our main rather general structural theorem points out that these categories arise in a rather natural mathematical way from the "linear maps" of Cartesian storage categories. As Cartesian storage categories have a rather natural mathematical genus, this helps to explain why "Seely categories" are such fundamental structures.

Our examination of the coKleisli categories of differential categories had led us to introduce the notion of a Cartesian differential category, [BCS09]. Cartesian differential categories are categories of "raw" differentiable maps with all the indirection—alluded to above—removed. Pleasingly these categories have a wide range of models. In particular, in [BCS09], we showed that the coKleisli category of a (monoidal) differential category—satisfying an additional interchange axiom—*is* a Cartesian differential category.

However, this had left open the issue of providing a converse for this result. In general, it is certainly *not* the case that a Cartesian differential category is the coKleisli

category of a (monoidal) differential category—a basic counterexample is the category of finite dimensional real vector spaces with smooth maps. Thus, it is natural to ask what extra conditions are required on a Cartesian differential category to make it a coKleisli category of a (monoidal) differential category. This paper provides a rather natural (and appealing) answer to this question. However, lest the reader think we have completely resolved this question, we hasten to admit that it still does not answer the question in complete generality. Furthermore, it leaves some other rather natural questions open: for example it is natural to wonder whether *every* Cartesian differential category arises as a full subcategory of the coKleisli category of a (monoidal) differential category.

The motivation for developing Cartesian storage categories[1] was that they provided an intermediate description of the coKleisli category of a (monoidal) differential category. These coKleisli categories were already proven to be Cartesian differential categories and, as such, had an obvious subcategory of linear maps—namely those which were linear in the natural differential sense. A Cartesian storage category uses the abstract properties of "linear maps" as the basis for its definition and is always the coKleisli category of its linear maps. In order for such a storage category to be the coKleisli category of a "Seely category", a further ingredient was required: one must be able to represent bilinear maps with a monoidal product. The theory of these categories—which we have taken to thinking of as the theory of storage—provides the theoretical core of the paper and is described in Sections 2 and 3.

In Section 4 we apply this theory to Cartesian differential categories. Cartesian differential categories exhibit a notable further coincidence of structure. When one can classify the linear maps of a Cartesian differential category one also automatically has a transformation called a "codereliction". Having a codereliction in any Cartesian storage category implies—assuming linear idempotents split—that one also has, for free, monoidal representation. Thus, to apply the theory of storage categories to Cartesian differential categories it suffices to know just that the linear maps are classified. Finally, but crucially for the coherence of the story, the linear maps then, with no further structural requirement, form a monoidal differential category. This provides the main result of the paper Theorem 4.3.1. Completing the circle then, at the end of the paper, we are in a position to point out that a closed Cartesian differential storage category is a model of Ehrhard and Regnier's differential λ-calculus [ER04].

Considering that we have been discussing two types of differential category, it seemed a good idea to qualify each with an appropriate adjective, so as to distinguish between them in a balanced manner. So we shall (in this paper) refer to differential categories as "monoidal differential categories", to contrast them with "Cartesian differential categories". In parallel with this, we introduce other terminological pairings which are natural from the perspective of this paper, but are also structures which have received attention under different names: For example, "monoidal storage categories" are essentially "Seely

[1]In [Kock12] a closely related theory, based on commutative monads, is developed. Interestingly that work was motivated by probability distributions and so is related to abstracting integration rather than differentiation.

categories" and "monoidal differential storage categories" are essentially what previously we called differential storage categories. Our purpose is not to precipitate any renaming of familiar structures but rather to emphasize the structural relationships which emerge from the story we tell here.

2. Cartesian Storage categories

In the discussion above two situations were described in which we selected from a Cartesian category certain maps, namely the coherent maps in stable domains and the linear maps among differentiable maps, and then used a classification of these maps to extract a Seely category. The aim of the next two sections is to show why this works as a general procedure and, moreover, can produce a monoidal storage (or "Seely") category. In fact, our aim is to prove much more: namely that *all* "Seely categories" (with an exact modality) arise in this manner.

We start by defining what is meant by a system $\mathcal{L}$ of linear maps. One way to view a linear system is as a subfibration of the simple fibration. In a category equipped with a system of linear maps, we shall often suppress explicit mention of $\mathcal{L}$, referring to maps belonging to this system as simply being *linear*. Such systems arise very naturally in Cartesian closed categories from exponentiable systems of maps and we make a detour to describe these connections.

Our initial aim is to characterize the categories in which there is a decomposition of maps analogous to that in linear logic. This eventually leads us to "Seely categories" as those in which all the "first-order" structural components of the logic are present.

Leaving definitions for later, the path will be the following: a "Cartesian storage category" is a category equipped with a system of linear maps which are "strongly and persistently classified". We then prove that this is equivalent to demanding that the category is a "strong abstract coKleisli category". This, in turn, is equivalent to being the coKleisli category of a category with a "forceful" comonad. Finally, to this story we add, in the next section, "monoidal representation", which when present, ensures the linear maps forms a monoidal storage category—this is essentially a "Seely category".

2.1. Systems of linear maps. By a **system of maps** of a category is meant a subcategory on the same objects: a **Cartesian** system of maps of a Cartesian category is just such a subcategory closed under the product structure. This means that projections, diagonal maps, and final map must be in the system and, furthermore, the pairing of any two maps in the system must be in the system. For a system $\mathcal{L}$ of maps to be linear it must determine a subfibration of the simple fibration.

The simple slice $\mathbb{X}[A]$ at A—that is the fiber over A of the simple fibration—of a Cartesian category $\mathbb{X}$, has the same objects as $\mathbb{X}$ but the homsets are modified $\mathbb{X}[A](X,Y) = \mathbb{X}(A \times X, Y)$. The composition in $\mathbb{X}[A]$ of two maps $g\colon A \times X \to Y$ and $h\colon A \times Y \to Z$ is $\langle \pi_0, g\rangle h$ with identities given by projection $\pi_1\colon A \times X \to X$. The substitution functor between simple slices has the following effect on the maps of $\mathbb{X}$. If $f\colon A \to B$ is

any map of $\mathbb{X}$, $g\colon X \to Y$ a map of $\mathbb{X}[B]$, (in $\mathbb{X}$ this is a map $g\colon B \times X \to Y$), then $\mathbb{X}[f](g) = (f \times 1)g\colon A \times X \to Y$, which is a map of $\mathbb{X}[A]$.

2.1.1. Definition. *A Cartesian category $\mathbb{X}$ is said to have a* **system of linear maps**, *$\mathcal{L}$, in case in each simple slice $\mathbb{X}[A]$ there is a system of maps $\mathcal{L}[A] \subseteq \mathbb{X}[A]$, which we shall refer to as the $\mathcal{L}[A]$-***linear** *maps (or, when $\mathcal{L}$ is understood, simply as "linear" maps), satisfying:*

[LS.1] *for each $X, Y, Z \in \mathbb{X}[A]$, all of $1_X, \pi_0\colon X \times Y \to X, \pi_1\colon X \times Y \to Y$ are in $\mathcal{L}[A]$ and, furthermore, $\langle f, g\rangle\colon X \to Y \times Z \in \mathcal{L}[A]$ whenever $f, g \in \mathcal{L}[A]$;*

[LS.2] *in each $\mathbb{X}[A]$, $\mathcal{L}[A]$, as a system of maps, is closed under composition, furthermore, whenever $g \in \mathcal{L}[A]$ is a retraction and $gh \in \mathcal{L}[A]$ then $h \in \mathcal{L}[A]$;*

[LS.3] *all substitution functors*

$$\frac{A \xrightarrow{f} B}{\mathbb{X}[B] \xrightarrow[\mathbb{X}[f]]{} \mathbb{X}[A]}$$

preserve linear maps.

Notice how a system of linear maps determines a subfibration:

$$\begin{tikzcd}
\mathcal{L}[\mathbb{X}] \arrow[r, "\mathcal{I}[\mathbb{X}]"] \arrow[d, "\partial_{\mathcal{L}}"] & \mathbb{X}[\mathbb{X}] \arrow[d, "\partial"] \\
\mathcal{L} \arrow[r, "\mathcal{I}"'] & \mathbb{X}
\end{tikzcd}$$

where $\mathcal{I}$ is the obvious inclusion and $\mathcal{L}[\mathbb{X}]$ is the category:

[Objects:] $(A, X) \in \mathbb{X}_0 \times \mathbb{X}_0$

[Maps:] $(f, g)\colon (A, X) \to (B, Y)$ where $g\colon A \to B$ is any map and $f\colon A \times X \to Y$ is linear in its second argument.

Composition and identities are as in $\mathbb{X}[\mathbb{X}]$, where the maps are the same but with no linear restriction. Composition is defined by $(f, g)(f', g') = (\langle \pi_0 g, f\rangle f', gg')$ and $(\pi_1, 1)$ is the identity map. Cartesian maps are of the form $(\pi_1, g)\colon (A, X) \to (B, X)$.

Furthermore we observe using **[LS.2]**:

2.1.2. Lemma. *For any $\mathbb{X}$ with a system of linear maps in any slice $\mathbb{X}[A]$:*

(i) if f is an linear map which is an isomorphism, then f^{-1} is linear;

(ii) if e is a linear idempotent which splits as a retraction r and a section i, and if r is linear, then i is also linear.

We shall need terminology which focuses on the linear argument rather than the context of that argument. Thus to say that a map $f\colon X \times A \to Y$ is linear *in its second argument* is equivalent to saying that f is $\mathcal{L}[X]$-linear. Similarly, a map, f, is linear *in its first argument* if $c_\times f$ is $\mathcal{L}[X]$-linear.

2.1.3. Lemma. *If $\mathcal{L}$ is a system of linear maps on a Cartesian category $\mathbb{X}$ then if $f\colon X \times A \to B$ and $g\colon Y \times B \to C$ are linear in their second arguments then $(f \times 1)g\colon Y \times X \times A \to C$ is linear in its third argument.*

Proof. As f is linear in $\mathbb{X}[X]$, $\mathbb{X}[\pi_1](f)\colon Y \times X \times A \to C$ is linear in $\mathbb{X}[Y \times X]$. Similarly as g is linear in $\mathbb{X}[Y]$, $\mathbb{X}[\pi_0](g)$ is linear in $\mathbb{X}[Y \times X]$. Now $(1 \times f)g = \mathbb{X}[\pi_1](f)\mathbb{X}[\pi_0](g)$ which makes it a composite of linear maps and so itself linear. ∎

The lemma means that to determine whether a composite is linear in a particular argument it suffices to know that each composition was at a linear argument.

If $\mathcal{L}$ is a system of linear maps then $f\colon A \times B \to C$ is ($\mathcal{L}$-)**bilinear** in case it is linear in each argument individually.

Finally we make the useful if immediate observation:

2.1.4. Lemma. *If $\mathbb{X}$ has a system of linear maps $\mathcal{L}$ then each simple slice, $\mathbb{X}[A]$, inherits a system of linear maps $\mathcal{L}[A]$.*

2.2. Closed systems of maps. A system of linear maps induces a system of maps on $\mathbb{X}$ itself, as $\mathbb{X} \cong \mathbb{X}[1]$. This class, which we shall denote by $\mathcal{L}[]$, will determine the whole system of linear maps when the category is Cartesian closed. However, clearly such an $\mathcal{L}[]$ must satisfy some special properties which we now outline.

2.2.1. Definition. *A system of maps $\mathcal{C} \subseteq \mathbb{X}$ is an* **exponentiable system of maps** *in case:*

[ES.1] *for each $X, Y, Z \in \mathbb{X}$, all of $1_X, \pi_0\colon X \times Y \to X, \pi_1\colon X \times Y \to Y, \Delta\colon X \to X \times X$ are in $\mathcal{C}$ and, furthermore, $\langle f, g\rangle\colon X \to Y \times Z \in \mathcal{C}$ whenever $f, g \in \mathcal{C}$;*

[ES.2] *$\mathcal{C}$ as a system of maps is closed under composition, moreover, whenever $g \in \mathcal{C}$ is a retraction and $gh \in \mathcal{C}$ then $h \in \mathcal{C}$;*

[ES.3] *whenever $f\colon A \to B$ is in $\mathcal{C}$ and $g\colon X \to Y$ is any map, then $g \Rightarrow f\colon Y \Rightarrow A \to X \Rightarrow B$ is in $\mathcal{C}$;*

[ES.4] *furthermore*

$$\begin{aligned} \eta[X] &= \mathrm{curry}(\pi_1)\colon A \to X \Rightarrow A \\ \mu[X] &= \mathrm{curry}((\Delta \times 1)(1 \times \mathrm{eval})\mathrm{eval})\colon X \Rightarrow (X \Rightarrow A) \to X \Rightarrow A \end{aligned}$$

are all in $\mathcal{C}$.

Here we use the exponential correspondence in the following form:

$$\frac{A \times X \xrightarrow{f} B}{X \xrightarrow[\mathrm{curry}(f)]{} A \Rightarrow B.}$$

2.2.2. Definition. *We shall say that a system of linear maps in a Cartesian closed category is* **closed** *in case*

- *each evaluation map is linear in its second (higher-order) argument.*
- *the system is closed under currying, that is if* $f\colon A \times B \times C \longrightarrow D$ *is linear in its second argument (B) then* $\mathrm{curry}(f)\colon B \times C \longrightarrow A \Rightarrow D$ *is linear in its first argument.*

Equivalently, when $\mathbb{X}$ *is Cartesian closed, a system of linear maps is closed provided* $k\colon B \times X \longrightarrow Y \in \mathcal{L}[B]$ *if and only if* $\mathrm{curry}(k) \in \mathcal{L}[]$.

2.2.3. Proposition. *In a Cartesian closed category* $\mathbb{X}$*, exponentiable systems of maps are in bijective correspondence with closed systems of linear maps.*

Proof. Given an exponentiable system of maps $\mathcal{C}$ we define a system of linear maps $\widetilde{\mathcal{C}}$, by saying $f\colon A \times B \longrightarrow C$ is linear in $\mathbb{X}[A]$ if and only if $\mathrm{curry}(f)\colon B \longrightarrow A \Rightarrow C$ is in $\mathcal{C}$. This defines a system of linear maps since:

[LS.1] In $\mathbb{X}[A]$ the identity map is $\pi_1\colon A \times X \longrightarrow X$ which is linear as $\mathrm{curry}(\pi_1)$ is. For the remainder note that $A \times X \xrightarrow{\pi_1} X \xrightarrow{f} Y$ has $\mathrm{curry}(\pi_1 f) = \mathrm{curry}(\pi_0)A \Rightarrow f$ so that whenever f is in $\mathcal{C}$ then $\mathrm{curry}(\pi_1 f)$ will be in $\mathcal{C}$. Finally note that $\mathrm{curry}(\langle f, g\rangle) = \langle \mathrm{curry}(f), \mathrm{curry}(g)\rangle(\langle A \Rightarrow \pi_0, A \Rightarrow \pi_1\rangle)^{-1}$.

[LS.2] The curry of a composite in $\mathbb{X}[B]$ is given by

$$\mathrm{curry}(\langle f, \pi_1\rangle g) = \mathrm{curry}(f)(B \Rightarrow \mathrm{curry}(g))\mu[B]$$

which is a composite of $\mathcal{C}$-maps when f and g are.

Suppose now that $g \bullet h$ in $\mathbb{X}[B]$ and $\mathrm{curry}(g), \mathrm{curry}(gh) \in \mathcal{C}$ with g a retraction (with section g^s) then $(B \Rightarrow \mathrm{curry}(g^s)\mu[B]$ is right inverse to $(B \Rightarrow \mathrm{curry}(g))\mu[B]$ where the latter is in $\mathcal{C}$. Also

$$B \Rightarrow \mathrm{curry}(gh))\mu[B] = B \Rightarrow \mathrm{curry}(g))\mu[B] B \Rightarrow \mathrm{curry}(h))\mu[B]$$

so that $B \Rightarrow \mathrm{curry}(h))\mu[B] \in \mathcal{C}$ But this means $\eta[B](B \Rightarrow \mathrm{curry}(h))\mu[B] = \mathrm{curry}(h)$ is in $\mathcal{C}$.

[LS.3] If $\mathrm{curry}(h) \in \mathcal{C}$ then $\mathrm{curry}(\mathbb{X}[f](h)) = \mathrm{curry}((1 \times f)h) = (f \Rightarrow B)\mathrm{curry}(h)$ so that the substitution functors preserve $\widetilde{\mathcal{C}}$-maps.

We observe that as $\mathrm{curry}(\mathrm{eval}) = 1$ that in $\widetilde{\mathcal{C}}$ the evaluation map must necessarily be linear in its second (higher-order) argument.

For the converse, given a system of linear maps $\mathcal{L}$ in which evaluation maps are linear in their second argument, we now show that $\mathcal{L}[]$ is an exponential system of maps. The only difficulties are **[ES.3]** and **[ES.4]**. For the former note that $f \Rightarrow g$ is obtained by currying:

$$A \times (A' \Rightarrow B') \xrightarrow{f \times 1} A' \times (A' \Rightarrow B') \xrightarrow{\mathrm{eval}} B' \xrightarrow{g} B$$

However observe that this is linear in its second argument so that currying preserves this establishing **[ES.3]**. For the latter the argument is similar as both η and μ are obtained by currying maps which are linear in the appropriate coordinates.

Finally we must argue that these transitions are inverse: that $\widetilde{\mathcal{C}}[] = \mathcal{C}$ is immediate. We show that $\widetilde{\mathcal{L}}[] = \mathcal{L}$ by showing $\mathcal{L} \subseteq \widetilde{\mathcal{L}}[]$ and the converse. Suppose $f \in \mathcal{L}[A]$ then $\mathrm{curry}(f) \in \mathcal{L}[]$ so that $f \in \widetilde{\mathcal{L}}[][A]$ as required. Conversely suppose $f \in \widetilde{\mathcal{L}}[][A]$ then $\mathrm{curry}(f) \in \mathcal{L}[]$ but this means $f = (1 \times \mathrm{curry}(f))\mathrm{eval}$ is in $\mathcal{L}[A]$. ∎

We note that when $\mathcal{L}$ is an exponentiable system of linear maps, the notion of being ($\mathcal{L}$-)bilinear becomes the requirement that both $\mathrm{curry}(f)$ and $\mathrm{curry}(c_\times f)$ are linear in $\mathbb{X}$.

2.3. Storage. A **Cartesian storage category** is a Cartesian category $\mathbb{X}$ together with a system of linear maps, $\mathcal{L}$, which is persistently and strongly classified,[2] in the following sense.

2.3.1. Definition.

(i) A system of linear maps is **classified** *if there is an object function S and a family of maps $\varphi_X \colon X \to S(X)$ such that for each $f \colon X \to Y$, there is a unique linear map $f^\sharp$ such that*

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\varphi_X"'] & Y \\ S(X) \arrow[ur, dotted, "f^\sharp"'] & \end{tikzcd}$$

commutes.

(ii) A system of linear maps is **strongly classified** *if there is an object function S and maps $X \xrightarrow{\varphi_X} S(X)$ such that for every $f \colon A \times X \to Y$ there is a unique $f^\sharp \colon A \times S(X) \to Y$ in $\mathcal{L}[A]$ (that is, $f^\sharp$ is linear in its second argument) making*

$$\begin{tikzcd} A \times X \arrow[r, "f"] \arrow[d, "1\times\varphi_X"'] & Y \\ A \times S(X) \arrow[ur, dotted, "f^\sharp"'] & \end{tikzcd}$$

commute.

(iii) A strong classification is said to be **persistent** *in case whenever $f \colon A \times B \times X \to Y$ is linear in its second argument B then $f^\sharp \colon A \times B \times S(X) \to Y$ is also linear in its second argument.*

[2]The notion of classification might (by comparison with the notion of a subobject classifier in a topos, for example) be called coclassification—we shall not do that. Furthermore, we also use the term "represented", especially in the sense that under the classification given by S and φ, $f^\sharp$ represents f.

2.3.2. Remark. When the linear maps are classified this makes the inclusion of the linear maps into a right adjoint. A strong classification makes the inclusion of the linear maps into the simple fibration a fibred right adjoint. A strong persistent classification allows a powerful proof technique which we shall use extensively in what follows. To establish the equality of

$$f, g\colon A \times \underbrace{S(X) \times \ldots S(X)}_{n} \to Y$$

when f and g are maps which are linear *individually* in their last n arguments, it suffices to show $(1 \times \varphi^n)f = (1 \times \varphi^n)g$. Note that the linearity precondition is vital!

Strong classification gives a morphism of fibrations defined by:

$$S[\mathbb{X}]\colon \mathbb{X}[\mathbb{X}] \to \mathcal{L}[\mathbb{X}]; (f, g) \mapsto ((f\varphi)^\sharp, g).$$

This is a functor on the total category. Identity maps are preserved because

$$\begin{tikzcd}
A \times X \arrow[r, "\pi_1"] \arrow[d, "1\times\varphi"'] & X \arrow[r, "\varphi"] & S(X) \\
A \times S(X) \arrow[urr, "\pi_1"'] & &
\end{tikzcd}$$

commutes making $(\pi_1\varphi)^\sharp = \pi_1$. Composition is preserved as

$$\begin{tikzcd}
A \times X \arrow[r, "\langle\pi_0 g, f\rangle"] \arrow[d, "1\times\varphi"'] & B \times Y \arrow[r, "f'"] \arrow[d, "1\times\varphi"] & Z \arrow[r, "\varphi"] & S(Z) \\
A \times S(X) \arrow[r, "\langle\pi_0 g, (f\varphi)^\sharp\rangle"'] & B \times S(Y) \arrow[urr, "(f'\varphi)^\sharp"'] & &
\end{tikzcd}$$

Finally $S[\mathbb{X}]$ preserves Cartesian arrows as $S[\mathbb{X}](\pi_1, g) = (\pi_1, g)\colon (A, S(X)) \to (B, S(X))$.

Therefore the classification is the couniversal property for the inclusion of the linear maps into the simple fibration, and so S is a fibred right adjoint.

2.3.3. Remark. We recall some definitions dealing with the notion of strength.

A **functor** $S\colon \mathbb{X} \to \mathbb{Y}$ between monoidal categories is **strong** if there is a natural transformation ("strength") $\theta^S\colon X \otimes S(Y) \to S(X \otimes Y)$ so that the following diagrams commute:

$$\begin{tikzcd}
\top \otimes S(Y) \arrow[rr, "\theta"] \arrow[dr, "u^L_\otimes"'] & & S(\top \otimes Y) \arrow[dl, "S(u^L_\otimes)"] \\
& S(Y) &
\end{tikzcd}
\qquad
\begin{tikzcd}
X \otimes (Y \otimes S(Z)) \arrow[r, "a_\otimes"] \arrow[d, "1\otimes\theta"'] & (X \otimes Y) \otimes S(Z)) \arrow[dd, "\theta"] \\
X \otimes S(Y \otimes Z) \arrow[d, "\theta"'] & \\
S(X \otimes (Y \otimes Z)) \arrow[r, "S(a_\otimes)"'] & S((X \otimes Y) \otimes Z)
\end{tikzcd}$$

The identity functor is strong (with strength given by the identity), and if S and T are strong, so is their composite, with strength given by $\theta^T T(\theta^S)\colon X \otimes T(S(Y)) \to T(X \otimes S(Y)) \to T(S(X \otimes Y))$.

A **natural transformation** $\psi\colon S \to T$ between strong functors is **strong** if the following commutes.

$$\begin{tikzcd} X \otimes S(Y) \arrow[r, "\theta^S"] \arrow[d, "1\otimes\psi"'] & S(X \otimes Y) \arrow[d, "\psi"] \\ X \otimes T(Y) \arrow[r, "\theta^T"'] & T(X \otimes Y) \end{tikzcd}$$

A **monad** (S, η, μ) is **strong** if each of S, η, and μ is strong.

Thus we have proved:

2.3.4. Lemma. *If $\mathbb{X}$ has a strongly classified system of linear maps, then $S\colon \mathbb{X} \to \mathbb{X}$ is a strong functor (and thus a morphism of simple fibrations) and determines a fibred right adjoint to the inclusion $\mathcal{I}\colon \partial_{\mathcal{L}} \to \partial$.*

The strength θ is given explicitly by

$$\begin{tikzcd} A \times X \arrow[r, "\varphi"] \arrow[d, "1\times\varphi"'] & S(A \times X) \\ A \times S(X) \arrow[ur, "\theta=\varphi^\sharp"'] & \end{tikzcd}$$

and all the natural transformations are appropriately strong. The strength of φ, the unit of the adjunction, is provided by the defining diagram of the strength transformation θ above. Notice that ϵ is *not* natural as ϵf is not necessarily linear unless f is; however, it does satisfy the strength requirement. However, $\mu = \epsilon_{S(_)}$ is natural and also strong.

This adjunction induces a comonad $\check{\mathcal{S}} = (S, \delta, \epsilon)$ on the linear maps whose data is given by

$$\begin{tikzcd} A \arrow[r, "\varphi"] \arrow[dr, equal] & S(A) \arrow[d, dotted, "\epsilon"] \\ & A \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "\varphi"] \arrow[d, "\varphi"'] & S(A) \arrow[d, dotted, "\delta=S(\varphi)"] \\ S(A) \arrow[r, "\varphi"'] & S(S(A)) \end{tikzcd}$$

Notice that by definition both ϵ and δ are linear (in $\mathcal{L}[]$).

In Cartesian storage categories the classification is also persistent, which means that when you make a function linear in an argument—by classifying it at that argument—you do not destroy the linearity that any of the other arguments might have enjoyed. This has the important consequence that the universal property can be extended to a multi-classification property as mentioned above. Here is the argument for the bi-classification property:

2.3.5. Lemma. *In a Cartesian storage category, for each $f\colon A \times X \times Y \longrightarrow Z$ there is a unique $f^{\sharp_2}$, which is linear in both its second and third arguments (*i.e. *bilinear), such that*

$$\begin{tikzcd} A \times X \times Y \arrow[r, "f"] \arrow[d, "1\times\varphi\times\varphi"'] & Z \\ A \times S(X) \times S(Y) \arrow[ur, dotted, "f^{\sharp_2}"'] & \end{tikzcd}$$

commutes.

Proof. To establish the existence of a map we may extend it in stages:

$$\begin{tikzcd} A \times X \times Y \arrow[r, "f"] \arrow[d, "1\times\varphi\times 1"'] & Z \\ A \times S(X) \times Y \arrow[ur, dotted, "f^{\sharp_1}"'] \arrow[d, "1\times 1\times\varphi"'] & \\ A \times S(X) \times S(Y) \arrow[uur, dotted, "f^{\sharp_2}"'] & \end{tikzcd}$$

the map $f^{\sharp_2}$ is then linear in its last two arguments by persistence. Suppose, for uniqueness, that a map g, which is linear in its last two arguments, has $(1 \times \varphi \times \varphi)g = f\colon A \times S(X) \times S(Y) \longrightarrow Z$. Then $f^{\sharp_1} = (1 \times 1 \times \varphi)g$ as the latter is certainly linear in its middle argument. Whence $g = f^{\sharp_2}$. ∎

This allows the observation that there is a candidate for monoidal structure $m_\times\colon S(A)\times S(B) \longrightarrow S(A \times B)$ given as the unique bilinear map lifting φ:

$$\begin{tikzcd} A \times B \arrow[r, "\varphi"] \arrow[d, "\varphi\times\varphi"'] & S(A \times B) \\ S(A) \times S(B) \arrow[ur, dotted, "m_\times"'] & \end{tikzcd}$$

We observe that this implies the following two identities:

2.3.6. Lemma. *In any Cartesian storage category:*

(i) $m_\times = \theta S(\theta')\mu$;

(ii) $(\epsilon \times \epsilon)m_\times = m_\times S(m_\times)\epsilon$.

Proof. Both can be verified using the universal property outlined in Lemma 2.3.5. First one notices that all the maps are bilinear, thus it suffices to show that prefixing with $\varphi\times\varphi$ gives the same maps:

(i) Clearly $m_\times$ is bilinear. Recall that $\mu = \epsilon_{S(_)}$. Notice that $\theta S(\theta')\mu$ is linear in its first argument as θ is and $S(\theta')\mu$ is linear. However, using persistence it is also linear in its second argument as:

$$\begin{tikzcd}
A \times S(B) \arrow[rr, "\theta'"] \arrow[d, "\varphi\times 1"'] \arrow[dr, "\varphi"] & & S(A \times B) \arrow[rr, equal] \arrow[dr, "\varphi"] & & S(A \times B) \\
S(A) \times S(B) \arrow[r, "\theta"'] & S(S(A) \times B) \arrow[rr, "S(\theta')"'] & & S^2(A \times B) \arrow[ur, "\epsilon_{S(A\times B)}=\mu"'] &
\end{tikzcd}$$

This gives:

$$\begin{aligned}
(\varphi \times \varphi)m_\times &= \varphi \\
(\varphi \times \varphi)\theta S(\theta')\mu &= (\varphi \times 1)(1 \times \varphi)\theta S(\theta')\mu \\
&= (\varphi \times 1)\varphi S(\theta')\mu \\
&= (\varphi \times 1)\theta'\varphi\mu \\
&= \varphi\varphi\mu = \varphi
\end{aligned}$$

(ii) In this case both maps are clearly bilinear and thus the equality is given by:

$$\begin{aligned}
(\varphi \times \varphi)(\epsilon \times \epsilon)m_\times &= m_\times \\
(\varphi \times \varphi)m_\times S(m_\times)\epsilon &= \varphi S(m_\times)\epsilon \\
&= m_\times\varphi\epsilon = m_\times
\end{aligned}$$

.

■

Then:

2.3.7. Proposition. *In a Cartesian storage category:*

(i) $\mathcal{S} = (S, \varphi, \mu)$ *is a commutative monad, that is* $\mathcal{S}$ *is a monoidal monad with respect to the product with* $m_\times = \theta S(\theta')\mu \colon S(A) \times S(B) \to S(A \times B)$*;*

(ii) for every object X

$$\begin{tikzcd}
S(S(X)) \arrow[r, shift left, "\epsilon"] \arrow[r, shift right, "S(\epsilon)"'] & S(X) \arrow[r, "\epsilon"] & X
\end{tikzcd}$$

is an absolute coequalizer;

(iii) $f \colon A \times X \to Y$ *is linear in its second argument if and only if*

$$\begin{tikzcd}
A \times S(X) \arrow[r, "\theta"] \arrow[d, "1\times\epsilon"'] & S(A \times X) \arrow[r, "S(f)"] & S(Y) \arrow[d, "\epsilon"] \\
A \times X \arrow[rr, "f"'] & & Y
\end{tikzcd}$$

commutes;

*(iv) and so $f\colon A \times B \longrightarrow Y$ is bilinear (*i.e. *linear in its first two arguments) if and only if*

$$\begin{tikzcd}
S(A) \times S(B) \arrow[r, "m_\times"] \arrow[d, "\epsilon\times\epsilon"'] & S(A \times B) \arrow[r, "S(f)"] & S(Y) \arrow[d, "\epsilon"] \\
A \times B \arrow[rr, "f"'] & & Y
\end{tikzcd}$$

commutes.

Proof.

(i) To establish this, given that we know the monad is strong, it suffices to show that the strength is commutative. That is that the following diagram commutes:

$$\begin{tikzcd}
S(A) \times S(B) \arrow[r, "\theta"] \arrow[d, "\theta'"'] & S(S(A) \times B) \arrow[r, "S(\theta')"] & S(S(A \times B)) \arrow[dd, "\mu"] \\
S(A \times S(B)) \arrow[d, "S(\theta)"'] & & \\
S(S(A \times B)) \arrow[rr, "\mu"'] & & S(A \times B)
\end{tikzcd}$$

The defining diagrams for the two routes round this square are:

$$\begin{tikzcd}
A \times B \arrow[r, "\varphi\times 1"] \arrow[dr, "\varphi"'] & S(A) \times B \arrow[r, "1\times\varphi"] \arrow[d, "\theta'"] \arrow[dr, "\varphi"] & S(A) \times S(B) \arrow[d, "\theta"] \\
& S(A \times B) \arrow[r, "\varphi"] \arrow[ddr, equal] & S(A \times S(B)) \arrow[d, "S(\theta')"] \\
& & S(S(A \times B)) \arrow[d, "\mu"] \\
& & S(A \times B)
\end{tikzcd}$$

and

$$\begin{tikzcd}
A \times B \arrow[r, "1\times\varphi"] \arrow[dr, "\varphi"'] & A \times S(B) \arrow[r, "\varphi\times 1"] \arrow[d, "\theta"] \arrow[dr, "\varphi"] & S(A) \times S(B) \arrow[d, "\theta'"] \\
& S(A \times B) \arrow[r, "\varphi"] \arrow[ddr, equal] & S(S(A) \times B) \arrow[d, "S(\theta)"] \\
& & S(S(A \times B)) \arrow[d, "\mu"] \\
& & S(A \times B)
\end{tikzcd}$$

Note that, because the classification is persistent, following Lemma 2.3.6 (i), both these constructed maps are bilinear. Thus, using the universal property of $\varphi \times \varphi$ they are equal. This shows that the two ways round the square making the strength commutative are equal.

(ii) ϵ is split by φ and so this is a split coequalizer: that is $\epsilon\varphi = \varphi S(\epsilon)$ and both $\varphi\epsilon = 1$ and $S(\varphi)\epsilon = 1$. Thus this is an absolute coequalizer.

(iii) If f is linear in its second argument then $(1\times\epsilon)f$ is also linear in its second argument (recall ϵ is linear). But this means $f^\sharp = (1\times\epsilon)f$. However, also we know $f^\sharp = \theta S(f)\epsilon$. Thus, the diagram (which shows linearity) commutes.

Conversely, suppose this diagram commutes; then notice it suffices to show that we can factor $f^\sharp = (1 \times \epsilon)g$. Then, as $1 \times \epsilon$ is linear and also a retraction, g must be linear. Finally g must of course be f as $f = (1 \times \varphi)f^\sharp = (1 \times \varphi)(1 \times \epsilon)g = g$. Now to show we can factor $f^\sharp$ in this manner, we can use exactness; it suffices to show that

$$A \times S^2(X) \underset{1 \times \epsilon}{\overset{1 \times S(\epsilon)}{\rightrightarrows}} A \times S(X) \xrightarrow{f^\sharp} Y$$

commutes. For this we have the following simple calculation which uses the fact that $\epsilon\epsilon = S(\epsilon)\epsilon$:

$$\begin{aligned}
(1 \times \epsilon)f^\sharp &= (1 \times \epsilon)\theta S(f)\epsilon \\
&= (1 \times \epsilon)(1 \times \epsilon)f = (1 \times \epsilon\epsilon)f \\
&= (1 \times S(\epsilon)\epsilon)f = (1 \times S(\epsilon))(1 \times \epsilon)f \\
&= (1 \times S(\epsilon))\theta S(f)\epsilon = (1 \times S(\epsilon))f^\sharp.
\end{aligned}$$

(iv) This is immediate from the preceding.

■

Notice that this means that in a Cartesian storage category, the system $\mathcal{L}$ may equivalently be viewed as being induced by the monad S. The second property says that the induced comonad on the subfibration of linear maps is an *exact modality*. In the next section, as we study this comonad in more detail, the significance of this property will become clear.

We end this subsection with two observations. The first of these is that the notion of a storage category is simply stable in the sense that each simple slice of a storage category is itself a storage category. This, in principle, will allow us to suppress the fibred context of a statement about storage categories, as such statements will be true in all simple-slice fibres.

2.3.8. Lemma. *If $\mathbb{X}$ is a Cartesian storage category then each simple slice, $\mathbb{X}[A]$, is also a Cartesian storage category.*

The second observation concerns the algebras of the monad $(S, \varphi, \mu = \epsilon_S)$. Recall that an algebra for this monad is an object A with a map $\nu\colon S(A) \to A$ such that

$$\begin{tikzcd} A \arrow[r, "\varphi"] \arrow[dr, equal] & S(A) \arrow[d, "\nu"] & A \arrow[l, "S(\varphi)"'] \arrow[dl, equal] \\ & A & \end{tikzcd} \qquad \begin{tikzcd} S^2(A) \arrow[r, "\epsilon_{S(A)}"] \arrow[d, "S(\nu)"'] & S(A) \arrow[d, "\nu"] \\ S(A) \arrow[r, "\nu"'] & A \end{tikzcd}$$

Consider the algebras whose structure maps are linear. Of course, there could be algebras whose structure map does not lie in $\mathcal{L}$ but for now we consider the case $\nu \in \mathcal{L}$. These algebras, due to the first triangle identity above, must be precisely of the form $\nu = \epsilon_A$— and then the remaining identities will automatically hold. A homomorphism of such algebras is then precisely a map $f\colon A \to B$ which is ϵ-natural. Thus, we have:

2.3.9. Lemma. *The full subcategory of S-algebras, determined by algebras whose structure maps is linear, is, for any Cartesian storage category $\mathbb{X}$, isomorphic to the subcategory of linear maps $\mathcal{L}[]$.*

2.4. Abstract coKleisli categories. Abstract coKleisli categories were introduced (in the dual) by Fuhrmann [F99] to provide a direct description of coKleisli categories. Thus, every coKleisli category is an abstract coKleisli category and, furthermore, from any abstract coKleisli category, $\mathbb{X}$, one can construct a subcategory, $\check{\mathbb{X}}$, with a comonad, $\check{\mathcal{S}}$, whose coKleisli category is exactly the original category $\mathbb{X}$. Of particular interest is when the constructed category, $\check{\mathbb{Y}}_{\mathcal{S}}$, of a coKleisli category $\mathbb{Y}_{\mathcal{S}}$ is precisely $\mathbb{Y}$: this, it turns out, happens precisely when $\mathcal{S}$ is an exact modality in the sense above.

This section views a Cartesian storage category from the perspective of its being an abstract coKleisli category. A Cartesian storage category is clearly a rather special abstract coKleisli category so that our aim is to determine what extra algebraic conditions are being required.

2.4.1. Definition. *An* **abstract coKleisli category** *is a category $\mathbb{X}$ equipped with a functor S, a natural transformation $\varphi\colon 1_{\mathbb{X}} \to S$, and a family of maps $\epsilon_A\colon S(A) \to A$ (which is not assumed to be natural) such that $\epsilon_{S(_)}$ is natural. Furthermore:*

$$\varphi\epsilon = 1, \quad S(\varphi)\epsilon = 1, \quad \textit{and} \quad \epsilon\epsilon = S(\epsilon)\epsilon$$

An abstract coKleisli category is said to be **Cartesian** *if its underlying category, $\mathbb{X}$, is a Cartesian category such that $!$, π_0 and π_1 are all ϵ-natural (in the sense that $\epsilon\pi_0 = S(\pi_0)\epsilon$,* etc.*).*

In an abstract coKleisli category the ϵ-natural maps, that is those f which satisfy $S(f)\epsilon = \epsilon f$, form a subcategory $\mathbb{X}_\epsilon$ and on this subcategory $\check{\mathcal{S}} = (S, S(\varphi), \epsilon)$ is a comonad.

On $\mathbb{X}$, the larger category, $\mathcal{S} = (S, \varphi, \epsilon_{S(_)})$ is a monad. It is not hard to see that the ϵ-natural maps form a system of linear maps which are classified:

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\varphi"'] & Y \\ S(X) \arrow[ur, "S(f)\epsilon"'] & \end{tikzcd}$$

where the uniqueness follows since if $h\colon S(X) \to Y$ has $\varphi h = f$ then $S(f)\epsilon = S(\varphi h)\epsilon = S(\varphi)\epsilon h = h$.

In an abstract coKleisli category, we shall always use the ϵ-natural maps as the default system $\mathcal{L}$ of maps. We now address the question of when this system is a linear system of maps.

2.4.2. Definition. *An abstract coKleisli category is said to be* **strong** *in case it is Cartesian and S is a strong functor, φ a strong natural transformation, and ϵ is strong, even where it is unnatural, meaning that the following commutes for all objects A and X:*

$$\begin{tikzcd} A \times S(X) \arrow[r, "\theta"] \arrow[dr, "1\times\epsilon"'] & S(A \times X) \arrow[d, "\epsilon"] \\ & A \times X \end{tikzcd}$$

In a strong abstract coKleisli category $\mathbb{X}$, $\mathcal{S} = (S, \varphi, \epsilon_{S(_)})$ is clearly a strong monad. We observe:

2.4.3. Proposition. *If $(\mathbb{X}, S, \varphi, \epsilon)$ is a strong abstract coKleisli category then the maps $f\colon A \times X \to Y$ of $\mathcal{L}[A] \subseteq \mathbb{X}[A]$ such that*

$$\begin{tikzcd} A \times S(X) \arrow[r, "\theta"] \arrow[d, "1\times\epsilon"'] & S(A \times X) \arrow[r, "S(f)"] & S(Y) \arrow[d, "\epsilon"] \\ A \times X \arrow[rr, "f"'] & & Y \end{tikzcd}$$

form a system, $\mathcal{L}_{\mathcal{S}}[\mathbb{X}]$, of linear maps which are strongly classified by (S, φ).

Proof. We check the conditions for being a system of linear maps:

[LS.1] The identity in $\mathbb{X}[A]$ is ϵ-natural since (by strength) $\theta; S(\pi_1) = \pi_1$:

$$\begin{tikzcd} A \times S(X) \arrow[r, "\theta"] \arrow[dr, "\pi_1"'] \arrow[dd, "1\times\epsilon"'] & S(A \times X) \arrow[d, "S(\pi_1)"] \\ & S(X) \arrow[d, "\epsilon"] \\ A \times X \arrow[r, "\pi_1"'] & X \end{tikzcd}$$

We check that the projections in $\mathbb{X}[A]$ are ϵ-natural using:

$$\begin{tikzcd}
A \times S(X_1 \times X_2) \arrow[r, "\theta"] \arrow[d, "1\times\epsilon"'] & S(A \times (X_1 \times X_2)) \arrow[r, "S(\pi_1)"] & S(X_1 \times X_2) \arrow[r, "S(\pi_i)"] \arrow[d, "\epsilon"] & S(X_i) \arrow[d, "\epsilon"] \\
A \times (X_1 \times X_2) \arrow[rr, "\pi_1"'] & & X_1 \times X_2 \arrow[r, "\pi_i"'] & X_i
\end{tikzcd}$$

Suppose $f_1\colon A \times X \to Y_1$ and $f_1\colon A \times X \to Y_1$ are ϵ-natural in $\mathbb{X}[A]$. Then in the following diagram the left hand square commutes if and only if the two outer squares corresponding to the two post compositions with the projections commute.

$$\begin{tikzcd}
A \times S(X) \arrow[r, "\theta"] \arrow[d, "1\times\epsilon"'] & S(A \times X) \arrow[r, "S(\langle f_1,f_2\rangle)"] & S(Y_1 \times Y_2) \arrow[r, "S(\pi_i)"] \arrow[d, "\epsilon"] & S(Y_i) \arrow[d, "\epsilon"] \\
A \times X \arrow[rr, "\langle f_1,f_2\rangle"'] & & Y_1 \times Y_2 \arrow[r, "\pi_i"'] & Y_i
\end{tikzcd}$$

However the outer squares commute by assumption and so $\langle f_1, f_2\rangle$ is ϵ-natural in $\mathbb{X}[A]$.

[LS.2] We must show that ϵ-natural maps in $\mathbb{X}[A]$ compose. We shall do it explicitly:

$$\begin{tikzcd}
 & S(A \times X) \arrow[r, "S(\Delta\times 1)"] \arrow[rrr, bend left, "S(\langle \pi_0,f\rangle)"] & S(A \times A \times X) \arrow[rr, "S(1\times f)"] & & S(A \times Y) \arrow[d, "S(g)"] \\
A \times S(X) \arrow[ur, "\theta"] \arrow[r, "\Delta\times 1"] \arrow[dd, "1\times\epsilon"'] & A \times A \times S(X) \arrow[r, "1\times\theta"] \arrow[dd, "1\times 1\times\epsilon"] & A \times S(A \times X) \arrow[u, "\theta"] \arrow[r, "1\times S(f)"] & A \times S(Y) \arrow[r, "\theta"] \arrow[dd, "1\times\epsilon"] & S(Z) \arrow[d, "\epsilon"] \\
 & & & & \\
A \times X \arrow[r, "\Delta\times 1"'] \arrow[rrr, bend right, "\langle \pi_0,f\rangle"'] & A \times A \times X \arrow[rr, "1\times f"'] & & A \times Y \arrow[r, "g"'] & Z
\end{tikzcd}$$

Next we must show that if gh is ϵ-natural and g is a retraction which is ϵ-natural then h is ϵ-natural. We shall prove it in the basic case leaving it as an exercise for the reader to do the calculation in a general slice.

We have

$$\begin{tikzcd} S(A) \arrow[r, "S(g)"] \arrow[d, "\epsilon"] & S(B) \arrow[r, "S(h)"] \arrow[d, "\epsilon"] & S(C) \arrow[d, "\epsilon"] \\ A \arrow[r, "g"] & B \arrow[r, "h"] & C \end{tikzcd}$$

where the left square is known to commute and the outer square commutes. This means $S(g)S(h)\epsilon = S(g)\epsilon h$ but $S(g)$ is a retraction as g is so the right square commutes.

[LS.3] We must show that if f is ϵ-natural in $\mathbb{X}[A]$ then $(g \times 1)f$ is ϵ-natural. Here is the diagram:

$$\begin{tikzcd} A \times S(X) \arrow[r, "\theta"] \arrow[dr, "g\times 1"] \arrow[dd, "\epsilon"] & S(A \times X) \arrow[rr, "S((g\times 1)f)"] \arrow[dr, "S(g\times 1)"] & & S(Y) \arrow[dd, "\epsilon"] \\ & B \times S(X) \arrow[r, "\theta"] \arrow[d, "1\times\epsilon"] & S(B \times X) \arrow[ur, "S(f)"] & \\ A \times X \arrow[r, "g\times 1"] & B \times X \arrow[rr, "f"] & & Y \end{tikzcd}$$

This system of linear maps is always strongly classified by (S, φ) by defining $f^\sharp = \theta S(f)\epsilon$, which is clearly ϵ-natural in $\mathbb{X}[A]$, and certainly has $(1 \times \varphi)\theta S(f)\epsilon = f$. It is unique, for if g is ϵ-natural in $\mathbb{X}[A]$ and has $(1 \times \varphi)g = f$, then

$$\theta S(f)\epsilon = \theta S((1 \times \varphi)g)\epsilon = (1 \times S(\varphi))\theta S(g)\epsilon = (1 \times S(\varphi))(1 \times \epsilon)g = g.$$

■

To complete the story it remains to establish the property to which persistence corresponds:

2.4.4. Lemma. *A strong abstract coKleisli category has a persistent strong classification of the ϵ-natural maps if and only if $\mathcal{S}$ is a commutative (or monoidal) monad.*

Proof. Suppose that $f\colon A \times X \times Y \to Z$ is linear in its second argument. That is

$$\begin{tikzcd} A \times S(X) \times Y \arrow[r, "\theta_2"] \arrow[d, "1\times\epsilon\times 1"] & S(A \times X \times Y) \arrow[r, "S(f)"] & S(Z) \arrow[d, "\epsilon"] \\ A \times X \times Y \arrow[rr, "f"] & & Z \end{tikzcd}$$

commutes. Then we must show that after linearizing the last argument the result is still linear in the second argument. That is the following diagram commutes:

$$\begin{tikzcd}
 & & S(A \times X \times S(Y)) \arrow[r, "S(\theta_3)"] & S^2(A \times X \times Y) \arrow[r, "S^2(f)"] \arrow[dr, "\epsilon"] & S^2(Z) \arrow[dr, "\epsilon"] & \\
A \times S(X) \times S(Y) \arrow[urr, "\theta_2"] \arrow[drr, "\theta_3"] \arrow[d, "1\times\epsilon\times 1"'] & & & & S(A \times X \times Y) \arrow[r, "S(f)"] & S(Z) \arrow[d, "\epsilon"] \\
A \times X \times S(Y) \arrow[drr, "\theta_3"'] & & S(A \times S(X) \times Y) \arrow[r, "S(\theta_2)"] \arrow[d, "S(1\times\epsilon\times 1)"] & S^2(A \times X \times Y) \arrow[ur, "\epsilon"] \arrow[r, "S^2(f)"'] & S^2(Z) \arrow[ur, "\epsilon"] \arrow[d, "S(\epsilon)"'] & Z \\
 & & S(A \times X \times Y) \arrow[rr, "S(f)"'] & & S(Z) \arrow[ur, "\epsilon"] &
\end{tikzcd}$$

Conversely we have already established that a persistent strong classification gives rise to a commutative monad. ∎

We shall say that a strong abstract coKleisli category is **commutative** if its monad is. We have now established:

2.4.5. Theorem. *If $\mathbb{X}$ is a Cartesian category, then $\mathbb{X}$ is a Cartesian storage category (that is, $\mathbb{X}$ has a persistently strongly classified system of linear maps) if and only if it is a strong abstract commutative coKleisli category.*

2.5. Linear idempotents. A very basic construction on a category is to split the idempotents and we pause our development to briefly consider how this construction applies to storage categories. In a storage category there is a slightly subtle issue as it is quite possible for a linear idempotent to split into two non-linear components. This means that even though an idempotent may split in the "large" category of all maps it need not in the subcategory of linear maps. We shall be interested in splitting the linear idempotents linearly, for in this case we can show that the resulting category is also a storage category which extends the original category without introducing any new linear maps.

We start with a basic observation:

2.5.1. Lemma. *In a Cartesian storage category:*

(i) When fg is linear and g is monic and linear, then f is linear;

(ii) When fg is linear and f is linear with $S(f)$ epic, then g is linear;

(iii) A linear idempotent $e\colon X \longrightarrow Y$ splits linearly, in the sense that there is a splitting (r, s) with $rs = e$ and $sr = 1$ with both r and s linear iff there is a splitting in which at least one of r and s are linear.

Proof. Consider

$$\begin{tikzcd}
S(X) \arrow[rr, "S(fg)"] \arrow[dr, "S(f)"] \arrow[dd, "\epsilon"'] & & S(Y) \arrow[dd, "\epsilon"] \\
& S(Y) \arrow[ur, "S(g)"'] \arrow[dd, "\epsilon"] & \\
X \arrow[rr, "fg"] \arrow[dr, "f"'] & & Y \\
& Y \arrow[ur, "g"'] &
\end{tikzcd}$$

Under the conditions of *(i)* the back face commutes since fg is linear (*i.e.* ϵ-natural), and the "g face" commutes. Then $S(f)\epsilon g = S(f)S(g)\epsilon = \epsilon fg$ and, thus, as g is monic, $S(f)\epsilon = \epsilon f$ implying f is linear.

Under the conditions of *(ii)* the back face still commutes and the "f face" commutes. So then $S(f)\epsilon g = \epsilon fg = S(g)F(g)\epsilon$ and, thus, as $S(f)$ is epic $\epsilon g = S(g)\epsilon$ implying g is linear.

Finally for *(iii)* if e splits and s is linear then $e = rs$ satisfies *(i)* while if r is linear it satisfies *(ii)* as $S(r)$ being a retraction is certainly epic. ∎

In a storage category we shall often wish to split all the linear idempotents, that is split all idempotents e such that $e \in \mathcal{L}$. An important observation is that this can be done entirely formally:

2.5.2. Proposition. *If $\mathbb{X}$ is a Cartesian storage category then the category obtained by splitting the linear idempotents, $\mathsf{Split}_{\mathcal{L}}(\mathbb{X})$, is a Cartesian storage category in which linear idempotents split and the embedding*

$$\mathcal{I}\colon \mathbb{X} \longrightarrow \mathsf{Split}_{\mathcal{L}}(\mathbb{X})$$

preserves and reflects linear maps and the classification.

Proof. The objects of $\mathsf{Split}_{\mathcal{L}}(\mathbb{X})$ are the linear idempotents in $\mathbb{X}$, and the maps $f\colon e \longrightarrow e'$ are, as usual, those such that $efe' = f$. As linear idempotents are closed under products (that is whenever e and e' are linear idempotents then $e \times e'$ is a linear idempotent) it is standard that these form a Cartesian category. We shall say that $f\colon e \times e' \longrightarrow e''$ is linear in its first argument precisely when it is so as a map in $\mathbb{X}$. It is then immediate that in $\mathsf{Split}_{\mathcal{L}}(\mathbb{X})$ linear idempotents will split linearly.

The functor S is now defined on $\mathsf{Split}_{\mathcal{L}}(\mathbb{X})$ by taking an object e to $S(e)$ and a map f to $S(f)$. The transformations φ and ϵ can also be extended by setting $\varphi_e := e\varphi S(e) = \varphi S(e) = e\varphi$ and $\epsilon_e := S(e)\epsilon e = \epsilon e = S(e)\epsilon$ (where here we use the linearity of e).

To show that (S, φ) on $\mathsf{Split}_{\mathcal{L}}(\mathbb{X})$ classifies linear maps in $\mathsf{Split}_{\mathcal{L}}(\mathbb{X})$ consider

$$\begin{tikzcd} a \times e \arrow[r, "f"] \arrow[d, "a\times\varphi"'] & e' \\ a \times S(e) \arrow[ur, dotted, "f^\sharp"'] & \end{tikzcd}$$

where a, e, e' are linear idempotents and $f = (a \times e)fe'$ and $f^\sharp$ is the linear classification for f in $\mathbb{X}$ then it suffices to show that $f^\sharp = (a \times S(e))f^\sharp e'$ to show this is a classification in $\mathsf{Split}_{\mathcal{L}}(\mathbb{X})$. To show this first note that $(a \times S(e))f^\sharp e'$ is linear, so if $(1 \times \varphi)(a \times S(e))f^\sharp e' = f$ we will have secured the equality. For this we note:

$$(1 \times \varphi)(a \times S(e))f^\sharp e' = (a \times e)(1 \times \varphi))f^\sharp e' = (a \times e)fe' = f.$$

Persistence now follows immediately from persistence in $\mathbb{X}$.

We also need to show that the linear maps in $\mathsf{Split}_{\mathcal{L}}(\mathbb{X})$ satisfy [**LS.1**], [**LS.2**] ,and [**LS.3**]. The only difficulty concerns [**LS.2**] and the condition on linear retractions. In a slice $\mathsf{Split}_{\mathcal{L}}(\mathbb{X})[a]$ we must show that if g is a linear retraction (where the section is not necessarily linear) and gh is linear then h is linear. This gives the following commuting diagram of maps

$$\begin{tikzcd} a \times e' \arrow[r, "\langle\pi_0, v\rangle"] \arrow[d, "a\times e'"'] & a \times e \arrow[d, "\langle\pi_0, g\rangle h"] \arrow[dl, "\langle\pi_0, g\rangle"] \\ a \times e' \arrow[r, "h"'] & e'' \end{tikzcd}$$

in which the downward pointing arrows contain the known linear components and v is the section of g which gives rise to the leftmost arrow being an "identity map" at the idempotent $a \times e'$. We must show that h is linear in this slice, that is that $(1 \times \epsilon)h = \theta S(h)\epsilon$. Here is the calculation:

$$\begin{aligned} \theta S(h)\epsilon &= \theta S((a \times e')h)\epsilon && \text{as } h\colon a \times e' \to e'' \\ &= \theta S(\langle\pi_0, v\rangle\langle\pi_0, g\rangle h)\epsilon && \text{as } v \text{ is a section of } g \\ &= \langle\pi_0, \theta S(v)\rangle\theta S(\langle\pi_0, g\rangle h)\epsilon \\ &= \langle\pi_0, \theta S(v)\rangle(1 \times \epsilon)\langle\pi_0, g\rangle h && \text{as } gh \text{ is linear} \\ &= \langle\pi_0, \theta S(\langle\pi_0, v\rangle g)\rangle(1 \times \epsilon)h && \text{as } g \text{ is linear} \\ &= \langle\pi_0, \theta S(\pi_1 e')\rangle(1 \times \epsilon)h && \text{as } v \text{ is a section of } g \\ &= (1 \times \epsilon)(1 \times e')h = (1 \times \epsilon)h. \end{aligned}$$

■

It is worth remarking that splitting linear idempotents of $\mathbb{X}$ does not cause linear idempotents in $\mathbb{X}[A]$ to split. One can, of course, split the linear idempotents in $\mathbb{X}[A]$ but this in general will have more objects than $\mathsf{Split}_{\mathcal{L}}(\mathbb{X})[A]$ as there will be more idempotents which are "linear in one argument" than are just linear. More precisely, a linear idempotent in $\mathbb{X}[A]$ is an $e'\colon A \times X \to X$ linear in X with $\langle\pi_0, e'\rangle e' = e'$ and these, in general, will strictly include idempotents $\pi_1 e$ where e is a linear idempotent in $\mathbb{X}$.

2.6. Forceful comonads. At this stage we have a fairly good grasp of what the monad of a Cartesian storage category looks like. However, notably absent has been a discussion of the properties of the comonad $\check{\mathcal{S}}$ on the linear maps of a Cartesian storage category. In this section we correct this defect.

Recall that the strength in a Cartesian storage category is not linear, in general, so that the comonad $\check{\mathcal{S}}$ will not have a strength. To guarantee that $\check{\mathcal{S}}$ be strong, we shall postulate the presence of a map which generates a strength map in the coKleisli category: this we call a *force* and we develop its properties below.

2.6.1. Definition. *A* **(commutative) force** *for a comonad* $\check{\mathcal{S}} = (S, \epsilon, \delta)$ *is a natural transformation*

$$S(A \times S(X)) \xrightarrow{\psi} S(A \times X)$$

which renders the following diagrams commutative:

[Force.1] *Associativity of force: ($a_\times$ is the associativity transformation)*

$$\begin{tikzcd}
S(A \times (B \times S(C)) \arrow[r, "a_\times"] \arrow[d, "\delta"'] & S((A \times B) \times S(C)) \arrow[ddd, "\psi"] \\
S^2(A \times (B \times S(C)) \arrow[d, "S(\sigma_\times)"'] & \\
S(S(A) \times S(B \times S(C)) \arrow[d, "S(\epsilon \times \psi)"'] & \\
S(A \times S(B \times C)) \arrow[d, "\psi"'] & \\
S(A \times (B \times C)) \arrow[r, "S(a_\times)"'] & S((A \times B) \times C)
\end{tikzcd}$$

[Force.2] *Projection and force:*

$$\begin{tikzcd}
S(A \times S(B)) \arrow[r, "\psi"] \arrow[d, "S(\pi_1)"'] & S(A \times B) \arrow[d, "S(\pi_1)"] \\
S^2(B) \arrow[r, "\epsilon"'] & S(B)
\end{tikzcd}$$

[Force.3] *Forceful naturality:*

$$\begin{tikzcd}
S(A \times S(B)) \arrow[r, "\delta"] \arrow[dd, "\psi"'] & S^2(A \times S(B)) \arrow[d, "S(\sigma_\times)"] \\
 & S(S(A) \times S^(B)) \arrow[d, "S(1\times(\epsilon\delta))"] \\
S(A \times B) \arrow[d, "\delta"'] & S(S(A) \times S^2(B)) \arrow[d, "\psi"] \\
S^2(A \times B) \arrow[r, "S(\sigma_\times)"'] & S(S(A) \times S(B))
\end{tikzcd}$$

[Force.4] *The forcefulness of counit:*

$$\begin{tikzcd}
S(A \times S^2(B)) \arrow[r, "S(1\times\epsilon)"] \arrow[d, "\psi"'] & S(A \times S(B)) \arrow[d, "\psi"] \\
S(A \times S(B)) \arrow[r, "\psi"'] & S(A \times B)
\end{tikzcd}$$

[Force.5] *The forcefulness of comultiplication:*

$$\begin{tikzcd}
S(A \times B) \arrow[r, "\delta"] \arrow[d, equal] & S^2(A \times B) \arrow[r, "S(\sigma_\times)"] & S(S(A) \times S(B)) \arrow[d, "\psi"] \\
S(A \times B) & & S(S(A) \times B) \arrow[ll, "S(\epsilon\times 1)"]
\end{tikzcd}$$

[Force.6] *Commutativity of force: ($c_\times$ is the commutativity transformation)*

$$\begin{tikzcd}
S(S(A) \times S(B)) \arrow[r, "\psi"] \arrow[d, "\psi'"'] & S(S(A) \times B) \arrow[d, "\psi'"] \\
S(A \times S(B)) \arrow[r, "\psi"'] & S(A \times B)
\end{tikzcd}$$

where $\psi' := S(c_\times)\psi S(c_\times)$ is the symmetric dual of the force.

We shall say that a comonad is **forceful** if it comes equipped with a force. First we observe the following.

2.6.2. Proposition. *In any Cartesian storage category $\mathbb{X}$ the comonad, $\check{\mathcal{S}}$ on the linear maps has a force.*

PROOF. If $\mathbb{X}$ is a Cartesian storage category we may define $\psi = S(\theta)\epsilon$: by inspection this is a linear map (*i.e.* in $\mathcal{L}[]$). We must check it satisfies all the given properties.

[Force.1] The interaction of associativity and strength gives the equation $(1\times\theta)\theta S(a_\times) = a_\times\theta$. We shall use this and the fact that $S(\varphi h)\epsilon = h$ for linear $h\colon S(X) \to Y$ in the calculation:

$$\begin{aligned}
\delta S(\sigma_\times)S(\epsilon\times\psi)\psi S(a_\times) &= S(\varphi\delta S(\sigma_\times)S(\epsilon\times\psi)\psi S(a_\times))\epsilon \\
&= S(\varphi\varphi S(\sigma_\times)S(\epsilon\times\psi)\psi S(a_\times))\epsilon \\
&= S(\varphi\sigma_\times(\epsilon\times\psi)\varphi\psi S(a_\times))\epsilon \\
&= S((\varphi\times\varphi)(\epsilon\times\psi)\varphi\psi S(a_\times))\epsilon \\
&= S((1\times\theta)\theta S(a_\times))\epsilon \\
&= S(a_\times\theta)\epsilon = S(a_\times)\psi
\end{aligned}$$

[Force.2] The projection of force:

$$\begin{aligned}
\psi S(\pi_1) &= S(\theta)\epsilon S(\pi_1) \\
&= S(\theta S(\pi_1))\epsilon \\
&= S(\pi_1)\epsilon
\end{aligned}$$

[Force.3] Forceful naturality:
We shall use the couniversal properties repeatedly: we need to show $\psi\delta S(\sigma_\times) = \delta S(\sigma_\times(1\times\epsilon\delta))\psi$, but as they are both linear, it suffices to show that they are equal when prefixed with φ:

$$\begin{aligned}
\varphi\psi\delta S(\sigma_\times) &= \theta\delta S(\sigma_\times) \\
&= (1\times\delta)\theta S(\theta\sigma_\times) \\
\varphi\delta S(\sigma_\times(1\times(\epsilon\delta)))\psi &= \varphi\varphi S(\sigma_\times(1\times(\epsilon\delta)))\psi \\
&= \varphi(\sigma_\times(1\times(\epsilon\delta)))\theta \\
&= (\varphi\times\varphi)(1\times(\epsilon\delta)))\theta \\
&= (\varphi\times\delta)\theta
\end{aligned}$$

The resulting expressions are both linear in the second argument, so they are equal provided precomposing with $1\times\varphi$ makes them equal:

$$\begin{aligned}
(1\times\varphi)(1\times\delta)\theta S(\theta\sigma_\times) &= (1\times(\varphi\varphi))\theta S(\theta\sigma_\times) \\
&= (1\times\varphi)\varphi S(\theta\sigma_\times) \\
&= (1\times\varphi)\theta\sigma_\times\varphi \\
&= \varphi\sigma_\times\varphi = (\varphi\times\varphi)\varphi \\
(1\times\varphi)(\varphi\times\delta)\theta &= (\varphi\times\varphi)(1\times\varphi)\theta \\
&= (\varphi\times\varphi)\varphi
\end{aligned}$$

This establishes the equality!

[Force.4] For the forcefulness of the counit both sides are linear as so equal if they are equal after precomposing with φ. However, $\varphi\psi\psi = \theta\psi$ and $\varphi S(1 \times \epsilon)\psi = (1 \times \epsilon)\varphi\psi = (1 \times \epsilon)\theta$. These last are linear in their second argument and, therefore, are equal if and only if precomposing with $1 \times \varphi$ makes them equal:

$$(1 \times \varphi)\theta\psi = \varphi\psi = \theta = (1 \times \varphi)(1 \times \epsilon)\theta$$

[Force.5] The forcefulness of the comultiplication:

$$\begin{aligned}
\delta S(\sigma_\times)\psi S(\epsilon \times 1) &= \delta S(\sigma_\times\theta)\epsilon S(\epsilon \times 1) \\
&= \delta S(\sigma_\times\theta S(\epsilon \times 1))\epsilon \\
&= S(\varphi\sigma_\times(\epsilon \times 1)\theta)\epsilon \\
&= S((\varphi \times \varphi)(\epsilon \times 1)\theta)\epsilon \\
&= S((1 \times \varphi)\theta)\epsilon \\
&= S(\varphi)\epsilon = 1
\end{aligned}$$

[Force.6] Commutativity of the force:

$$\begin{aligned}
\psi\psi' &= S(\theta)\epsilon S(\theta')\epsilon \\
&= S(\theta S(\theta')\epsilon\epsilon \\
&= S(\theta' S(\theta))\epsilon\epsilon \\
&= S(\theta')\epsilon S(\theta)\epsilon \\
&= \psi'\psi
\end{aligned}$$

■

Conversely:

2.6.3. Proposition. *Let $\mathcal{S}$ be a forceful comonad on $\mathbb{X}$ a Cartesian category then the coKleisli category $\mathbb{X}_{\mathcal{S}}$ is a Cartesian storage category.*

Proof. The coKleisli category is immediately an abstract coKleisli category (see [F99]): it remains only to show that all the data (S, φ, and ϵ) are strong, and that the induced monad is commutative.

The following is the interpretation for the data of the Cartesian storage category:

$$\begin{aligned}
[\![\varphi]\!] &:= 1_{S(A)} \\
[\![\epsilon]\!] &:= \epsilon\epsilon \\
[\![S(f)]\!] &:= \epsilon\delta S([\![f]\!]) \\
[\![\theta]\!] &:= \psi \\
[\![f \times g]\!] &:= \langle S(\pi_0)[\![f]\!], S(\pi_1)[\![g]\!]\rangle = \sigma_\times([\![f]\!] \times [\![g]\!]) \\
[\![\Delta]\!] &:= \epsilon\Delta \\
[\![\pi_i]\!] &:= \epsilon\pi_i \\
[\![a_\times]\!] &:= \epsilon a_\times \\
[\![fg]\!] &:= \delta S([\![f]\!])[\![g]\!].
\end{aligned}$$

First we must show that $\boldsymbol{\theta}$ satisfies the requirements of being a strength for $\mathbf{S}$ with respect to the product. This may be expressed as following two requirements and naturality:

$$\begin{tikzcd}
A \times S(X) \arrow[r, "\theta"] \arrow[dr, "\pi_1"'] & S(A \times X) \arrow[d, "S(\pi_1)"] \\
& S(X)
\end{tikzcd}
\qquad
\begin{tikzcd}
A \times (B \times S(X)) \arrow[r, "1\times\theta"] \arrow[d, "a_\times"'] & A \times S(B \times X) \arrow[r, "\theta"] & S(A \times (B \times X)) \arrow[d, "S(a_\times)"] \\
(A \times B) \times S(X) \arrow[rr, "\theta"'] & & S((A \times B) \times X)
\end{tikzcd}$$

The two diagrams are verified by the following calculation:

$$\begin{aligned}
[\![\theta \bullet S(\pi_1)]\!] &= \delta S(\psi)\epsilon\delta S(\epsilon\pi_1) \\
&= \psi S(\pi_1) = S(\pi_1)\epsilon = \epsilon\pi_1 \\
&= [\![\pi_1]\!] \\
[\![(1 \times \theta) \bullet \theta \bullet S(a_\times)]\!] &= \delta S(\langle S(\pi_0)\epsilon, S(\pi_1)\psi\rangle)\delta S(\psi)\epsilon\delta S(\epsilon a_\times) \\
&= \delta S(\langle S(\pi_0)\epsilon, S(\pi_1)\psi\rangle)\psi S(a_\times) \\
&= S(a_\times)\psi \\
&= [\![a_\times \bullet \theta]\!]
\end{aligned}$$

For naturality we have:

$$\begin{aligned}
[\![(f \times S(g))\theta]\!] &= \delta S(\sigma_\times(f \times (\epsilon\delta S(g))))\psi \\
&= \delta S(\sigma_\times(1 \times (\epsilon\delta)))S(f \times S(g))\psi \\
&= \delta S(\sigma_\times(1 \times (\epsilon\delta)))\psi S(f \times g) \\
&= \psi\delta S(\sigma_\times)S(f \times g) \\
&= \delta S(\psi)\epsilon\delta S(\sigma_\times(f \times g)) = [\![\theta S(f \times g)]\!]
\end{aligned}$$

Next we must verify that both $\boldsymbol{\varphi}$ and $\boldsymbol{\epsilon}_S$ (*i.e.* $\boldsymbol{\mu}$) are strong:

$$\begin{tikzcd} A \times X \arrow[r, "1\times\varphi"] \arrow[dr, "\varphi"'] & A \times S(X) \arrow[d, "\theta"] \\ & S(A \times X) \end{tikzcd} \qquad \begin{tikzcd} A \times S^2(X) \arrow[r, "\theta"] \arrow[d, "1\times\epsilon_S"'] & S(A \times S(X)) \arrow[r, "S(\theta)"] & S^2(A \times X) \arrow[d, "\epsilon_S"] \\ A \times S(X) \arrow[rr, "\theta"'] & & S(A \times X) \end{tikzcd}$$

$$\begin{aligned} [\![(1 \times \varphi)\theta]\!] &= \delta S(\sigma_\times(\epsilon \times 1))\psi \\ &= \delta S(\sigma_\times)\psi S(\epsilon \times 1) \\ &= 1 = [\![\varphi]\!] \\ [\![(1 \times \epsilon)\theta]\!] &= \delta S(\sigma_\times(\epsilon \times \epsilon\epsilon))\psi \\ &= \delta S(\sigma_\times(\epsilon \times \epsilon))S(1 \times \epsilon)\psi \\ &= \delta S(\sigma_\times(\epsilon \times \epsilon))\psi\psi \\ &= \psi\psi \\ &= \delta S(\psi\delta S(\psi)\epsilon)\epsilon \\ &= \delta S(\psi)\delta S(\epsilon\delta S(\psi))\epsilon\epsilon \\ &= [\![\theta S(\theta)\epsilon]\!] \end{aligned}$$

where we used the following equality $\delta S(\sigma_\times(\epsilon \times \epsilon)) = 1_{S(A\times B)}$ which holds as

$$\delta S(\sigma_\times(\epsilon \times \epsilon)) = \delta S(\langle S(\pi_0), S(\pi_1)\rangle(\epsilon \times \epsilon)) = \delta S(\langle \epsilon\pi_0, \epsilon\pi_1\rangle)) = \delta S(\epsilon\langle \pi_0, \pi_1\rangle) = \delta S(\epsilon) = 1.$$

Lastly we must show that the induced monad is commutative, that is:

$$\begin{tikzcd} S(A) \times S(B) \arrow[r, "\theta"] \arrow[d, "\theta'"'] & S(S(A) \times B) \arrow[r, "S(\theta')"] & S^2(A \times B) \arrow[d, "\mu"] \\ S(A \times S(B)) \arrow[r, "S(\theta)"'] & S^2(A \times B) \arrow[r, "\mu"'] & S(A \times B) \end{tikzcd}$$

$$\begin{aligned} [\![\theta S(\theta')\mu]\!] &= \delta S(\psi)(\delta S(\epsilon\delta S(\psi'))\epsilon\epsilon \\ &= \delta S(\psi\delta S(\psi')\epsilon)\epsilon \\ &= \psi\psi' = \psi'\psi \\ &= \delta S(\psi'\delta S(\psi)\epsilon)\epsilon \\ &= \delta S(\psi')(\delta S(\epsilon\delta S(\psi))\epsilon\epsilon \\ &= [\![\theta' S(\theta)\mu]\!] \end{aligned}$$

■

We shall call a Cartesian category with a forceful comonad a **Cartesian linear category**. We may summarize the above results as follows:

2.6.4. Theorem. *A category is a Cartesian storage category if and only if it is the coKleisli category of a Cartesian linear category. A Cartesian linear category is the linear maps of a Cartesian storage category if and only if its comonad is exact.*

Recall that a comonad is exact when the commuting diagram:

$$S(S(X)) \underset{\epsilon}{\overset{S(\epsilon)}{\rightrightarrows}} S(X) \xrightarrow{\epsilon} X$$

is a coequalizer. A category with an exact comonad is always the subcategory of ϵ-natural maps of its coKleisli category. This allows the original category to be completely recovered from the coKleisli category.

Starting with a forceful comonad $\mathcal{S}$ on a Cartesian category $\mathbb{X}$ one can directly form the simple fibration of the coKleisli category. The total category $\mathbb{X}$ can be described by:

Objects: (A, X) where $A, X \in \mathbb{Y}$;

Maps: $(f, g)\colon (A, X) \to (B, Y)$ where $f\colon S(A) \to B$ and $g\colon S(A \times X) \to Y$;

Identities: $(\epsilon, S(\pi_1)\epsilon)\colon (A, X) \to (A, X)$;

Composition: $(f, g)(f', g') = (\delta S(f)f', \delta S(\langle S(\pi_0)f, g\rangle)g')$.

That this amounts to the simple fibration $\partial\colon \mathbb{X}_{\mathcal{S}}[\mathbb{X}_{\mathcal{S}}] \to \mathbb{X}_{\mathcal{S}}; (f, g) \mapsto f$ is easily checked.

2.6.5. Remark. One might expect for there to be more interactions between the comultiplication and the force. In fact, there are but they are often not so obvious! Here is an example:

$$\begin{tikzcd}
S(A \times S(B)) \arrow[r, "S(1\times\delta)"] \arrow[d, "\psi"'] & S(A \times S^2(B)) \arrow[d, "\psi"] \\
S(A \times B) & S(A \times S(B)) \arrow[l, "\psi"]
\end{tikzcd}$$

An easy, but perhaps rather unsatisfactory way to check this is by looking in the corresponding Cartesian storage category where we have rather easily:

$$\begin{aligned}
S(1 \times \delta)\psi\psi &= S(1 \times S(\varphi))S(\theta)\epsilon S(\theta)\epsilon \\
&= S((1 \times S(\varphi))\theta S(\theta))\epsilon\epsilon \\
&= S(\theta S((1 \times \varphi)\theta)\epsilon)\epsilon \\
&= S(\theta\varphi\epsilon)\epsilon = S(\theta)\epsilon = \psi
\end{aligned}$$

However, one might reasonably want a direct proof. We shall use the following equality

$$\delta S(\sigma_\times(\epsilon \times \epsilon)) = 1_{S(A\times B)}$$

which holds since

$$\delta S(\sigma_\times(\epsilon \times \epsilon)) = \delta S(\langle S(\pi_0), S(\pi_1)\rangle(\epsilon \times \epsilon)) = \delta S(\langle \epsilon\pi_0, \epsilon\pi_1\rangle)) = \delta S(\epsilon\langle \pi_0, \pi_1\rangle) = \delta S(\epsilon) = 1.$$

Here is the calculation:

$$\begin{aligned}
\psi &= S(1 \times \delta S(\epsilon))\psi = (1 \times \delta)\psi S(1 \times \epsilon) \\
&= (1 \times \delta)\psi\delta S(\sigma_\times)\psi S(\epsilon \times 1)S(1 \times \epsilon) \\
&= (1 \times \delta)\psi\delta S(\sigma_\times)\psi S(\epsilon \times \epsilon) \\
&= (1 \times \delta)\delta S(\sigma_\times)S(1 \times (\epsilon\delta))\psi\psi S(\epsilon \times \epsilon) \\
&= (1 \times \delta)\delta S(\sigma_\times)S(1 \times (\epsilon\delta))\psi S(\epsilon \times S(\epsilon))\psi \\
&= (1 \times \delta)\delta S(\sigma_\times)S(1 \times (\epsilon\delta))S(\epsilon \times S^2(\epsilon))\psi\psi \\
&= (1 \times \delta)\delta S(\sigma_\times)S(\epsilon \times (\epsilon\delta S^2(\epsilon)))\psi\psi \\
&= (1 \times \delta)\delta S(\sigma_\times)(\epsilon \times \epsilon)(1 \times (S(\epsilon)\delta))\psi\psi \\
&= (1 \times \delta)(1 \times (S(\epsilon)\delta))\psi\psi \\
&= (1 \times \delta)\psi\psi
\end{aligned}$$

We now have the following which summarizes the main results of this section

2.6.6. Theorem. *For a Cartesian category $\mathbb{X}$, the following are equivalent:*

- $\mathbb{X}$ *is a Cartesian storage category;*
- $\mathbb{X}$ *is a strong abstract commutative coKleisli category;*
- $\mathbb{X}$ *is the coKleisli category of a Cartesian category with a forceful comonad.*

3. Monoidal storage categories

Our objective is now to link our development to the categorical semantics of linear logic through the "storage categories" which were developed from the ideas introduced in [S89]. The original definition of these categories required a comonad S on a $*$-autonomous category and natural isomorphisms $s_\times\colon S(A \times B) \to S(A) \otimes S(B)$ and $s_1\colon S(1) \to \top$. Subsequently Gavin Bierman realized that, in order to ensure that equivalent proofs of Multiplicative Exponential Linear Logic (MELL) were sent to the same map, the comonad actually had to be monoidal. Bierman called these categories, in which this new requirement was added, "new Seely" categories.

Bierman's examination of MELL also revealed that, even in the absence of additives, the (monoidal closure of the) coKleisli category of a symmetric monoidal category with an exponential would have products and, furthermore, when the original category was closed, would be Cartesian closed. Thus, it seemed that the additive structure was not really necessary for the theory. Bierman called the categories which provided models for

MELL *linear categories*. In this manner, the additive structure, basic in Seely's original definition, was relegated to a secondary and largely optional role.

Andrea Schalk [S04] collected the various axiomatizations of Seely categories originating from this work, and removed the requirement that the category be closed. She showed that this was an orthogonal property whose main purpose was to ensure the coKleisli category was Cartesian closed. She called the modalities *linear exponential comonads*, and thus replaced linear categories with symmetric monoidal categories with a linear exponential comonad.

Even more recently Paul-André Melliès [M02] while revisiting "categorical models of linear logic" concentrated entirely upon the exponential structure: notably $*$-autonomous categories hardly rated a mention and the additives are reduced to at most products—although closedness is assumed throughout. Of note was the emphasis that was placed on the role of the monoidal adjunction which was induced between the linear and Cartesian categories. Of particular interest was an axiomatization for Seely's original ideas which showed that rather than demand that the comonad be a monoidal comonad one could obtain the same effect by axiomatizing the Seely isomorphism itself more carefully. Here we shall follow this idea and, adapting it somewhat, obtain a convenient description of the variety of Seely category in which we are interested. These are "new Seely categories", but following Andrea Schalk's lead, the requirement of closedness is dropped.

One thing that should be emphasized is that in, this exposition of Seely categories, we are focusing on the product structure: the monoidal structure is regarded as secondary, indeed, even generated by products. This is absolutely the opposite to the general trend in the work cited above where an underlying theme is to decompose the product structure into a more fundamental monoidal structure. Mathematically, of course, there is no tension between these approaches, as in these settings the product and monoidal are linked and it should be no surprise that the linkage can be worked in both directions. However, there is perhaps a philosophical message, as it challenges the precept of what should be taken as primary. Seemingly in spite of this, we shall call the current notion "monoidal storage categories", as they are very similar to the "storage categories" we considered in [BCS06], in the context of (monoidal) differential categories.

The section starts by providing an exposition of monoidal storage categories: we start from the definition of a storage transformation and explain why this is the same as a coalgebra modality. A monoidal storage category is then defined to be a symmetric monoidal category with products and a storage transformation which is an isomorphism. We then prove that this implies that the modality is a monoidal comonad and, thus, that we do indeed obtain (modulo the relaxing of the closed requirement) what Gavin Bierman called a "new Seely" category.

Next, we return to the main theme of the paper, and formulate the notion of monoidal representation in a Cartesian storage category. We then show that the linear maps of a Cartesian storage category, which has persistent monoidal representation, always form a monoidal storage category. Conversely, the coKleisli category of a monoidal storage category is a Cartesian storage category with monoidal representation. Thus, monoidal

storage categories, in which the comonad is exact, correspond *precisely* to the linear maps of Cartesian storage categories which have persistent monoidal representation.

3.1. Coalgebra modalities. Let $\mathbb{X}$ be a symmetric monoidal category with tensor $(\otimes, a_\otimes, c_\otimes, u^L_\otimes, u^R_\otimes)$ which has products and a comonad (S, δ, ϵ). (Note that $a_\otimes, c_\otimes$ are associativity and commutativity iso-transformations for the tensor, and $u^L_\otimes, u^R_\otimes$ are the tensor unit iso-transformations.)

3.1.1. Definition. *A* **storage transformation** *is a symmetric comonoidal transformation $s\colon S \longrightarrow S$ from $\mathbb{X}$, regarded as a symmetric monoidal category with respect to the Cartesian product, to $\mathbb{X}$, regarded as a symmetric monoidal category with respect to the tensor, for which δ is a comonoidal transformation.*

Thus, a storage transformation is a natural transformation $s_2\colon S(X \times Y) \longrightarrow S(X) \otimes S(Y)$ and a map $s_1\colon S(1) \longrightarrow \top$ satisfying:

$$\begin{tikzcd}
S((X \times Y) \times Z) \arrow[r, "s_2"] \arrow[d, "S(a_\times)"'] & S(X \times Y) \otimes S(Z) \arrow[r, "s_2 \otimes 1"] & (S(X) \otimes S(Y)) \otimes S(Z) \arrow[d, "a_\otimes"] \\
S(X \times (Y \times Z)) \arrow[r, "s_2"'] & S(X) \otimes S(Y \times Z) \arrow[r, "1 \otimes s_2"'] & S(X) \otimes (S(Y) \otimes S(Z))
\end{tikzcd}$$

$$\begin{tikzcd}
S(1 \times X) \arrow[r, "s_2"] \arrow[d, "S(\pi_1)"'] & S(1) \otimes S(X) \arrow[d, "s_0 \otimes 1"] \\
S(X) & \top \otimes S(X) \arrow[l, "u^L_\otimes"]
\end{tikzcd}
\qquad
\begin{tikzcd}
S(X \times 1) \arrow[r, "s_2"] \arrow[d, "S(\pi_0)"'] & S(X) \otimes S(1) \arrow[d, "1 \otimes s_0"] \\
S(X) & S(X) \otimes \top \arrow[l, "u^R_\otimes"]
\end{tikzcd}$$

$$\begin{tikzcd}
S(X \times Y) \arrow[r, "s_2"] \arrow[d, "S(c_\times)"'] & S(X) \otimes S(Y) \arrow[d, "c_\otimes"] \\
S(Y \times X) \arrow[r, "s_2"'] & S(Y) \otimes S(X)
\end{tikzcd}$$

In general, a natural transformation is a comonoidal transformation if it respects the comonoidal structure in the sense that the following diagrams commute:

$$\begin{tikzcd}
F(X \times Y) \arrow[r, "\alpha"] \arrow[d, "\sigma^F_2"'] & G(X \times Y) \arrow[d, "\sigma^G_2"] \\
F(X) \otimes F(Y) \arrow[r, "\alpha \otimes \alpha"'] & G(X) \otimes G(Y)
\end{tikzcd}
\qquad
\begin{tikzcd}
F(1) \arrow[rr, "\alpha"] \arrow[dr, "\sigma^F_0"'] & & G(1) \arrow[dl, "\sigma^G_0"] \\
& \top &
\end{tikzcd}$$

A comonad is comonoidal if all its transformations are. However, for the comonad (S, δ, ϵ) it makes no sense to insist that ϵ is comonoidal as the identity functor is not comonoidal (from $\mathbb{X}$ with products to $\mathbb{X}$ with tensor). However, it does make sense to require δ is comonoidal. Recall first that, with respect to the product, every functor is canonically comonoidal with:

$$\sigma^\times_2 = \langle S(\pi_0), S(\pi_1) \rangle\colon S(X \times Y) \longrightarrow S(X) \times S(Y) \quad \text{and} \quad \sigma^\times_0 = \langle\rangle\colon S(1) \longrightarrow 1.$$

This allows us to express the requirement that δ be comonoidal as follows:

$$\begin{tikzcd}
S(X \times Y) \arrow[r, "\delta"] \arrow[dd, "s^2"'] & S(S(X \times Y)) \arrow[d, "\sigma_2^{\times}"] \\
 & S(S(X) \times S(Y)) \arrow[d, "s_2"] \\
S(X) \otimes S(Y) \arrow[r, "\delta\otimes\delta"'] & S(S(X)) \otimes S(S(Y))
\end{tikzcd}
\qquad
\begin{tikzcd}
S(1) \arrow[r, "\delta"] \arrow[dr, "s_0"'] & S(S(1)) \arrow[d, "S(\sigma_0^{\times})"] \\
 & S(1) \arrow[d, "s_0"] \\
 & \top
\end{tikzcd}$$

3.1.2. Definition. *A symmetric monoidal category $\mathbb{X}$ with products has a* **commutative coalgebra modality** *in case there is a comonad (S, δ, ϵ) such that each $S(X)$ is naturally a cocommutative comonoid $(S(X), \Delta, e)$ and δ is a homomorphism of these monoids:*

$$\begin{tikzcd}
S(X) \arrow[r, "\delta"] \arrow[d, "\Delta"'] & S(S(X)) \arrow[d, "\Delta"] \\
S(X) \otimes S(X) \arrow[r, "\delta\otimes\delta"'] & S(S(X)) \otimes S(S(X))
\end{tikzcd}
\qquad
\begin{tikzcd}
S(X) \arrow[rr, "\delta"] \arrow[dr, "e"'] & & S(S(X)) \arrow[dl, "e"] \\
 & \top &
\end{tikzcd}$$

We now observe that:

3.1.3. Proposition. *For a symmetric monoidal category with products, having a comonad with a symmetric storage transformation is equivalent to having a cocommutative coalgebra modality.*

Proof.

($\Rightarrow$) If one has a storage transformation then one can define natural transformations Δ, e as $\Delta = S(\Delta_{\times})s_2 \colon S(X) \to S(X) \otimes S(X)$ and $e = S(\langle\rangle)s_0 \colon S(X) \to \top$. As (symmetric) comonoidal functors preserve (commutative) comonoids these do define comonoids. Further, since δ is comonoidal as a transformation it becomes a homomorphism of the induced comonoids. This means that we have a (cocommutative) coalgebra modality.

($\Leftarrow$) Conversely given a (cocommutative) coalgebra modality on a (symmetric) monoidal category we may define $s_2 = \Delta(S(\pi_0) \otimes S(\pi_1)) \colon S(X \times Y) \to S(X) \otimes S(Y)$ and $s_0 = e \colon S(1) \to \top$. s_2 is clearly a natural transformation and using the fact that Δ is coassociative it is easily seen to satisfy the first comonoidal requirement. For the last we have:

$$s_2(1 \otimes s_0)u_{\otimes}^R = \Delta(S(\pi_0) \otimes S(\pi_1))(1 \otimes e)u_{\otimes}^R = \Delta(1) \otimes S(\pi_1))u_{\otimes}^R S(\pi_0) = S(\pi_0).$$

Thus these maps do provide comonoidal structure for S. It remains to show that δ is a comonoidal transformation:

$$\begin{aligned}
\delta S(\sigma_2^\times)s_2 &= \delta S(\sigma_2^\times)\Delta(S(\pi_0)\otimes S(\pi_1)) \\
&= \delta\Delta(S(\sigma_2^\times)\otimes S(\sigma_2^\times))(S(\pi_0)\otimes S(\pi_1)) \\
&= \delta\Delta(S(\sigma_2^\times\pi_0)\otimes S(\sigma_2^\times\pi_1)) \\
&= \delta\Delta(S(S(\pi_0))\otimes S(S(\pi_)) \\
&= \Delta(\delta\otimes\delta)(S(S(\pi_0))\otimes S(S(\pi_)) \\
&= \Delta(S(\pi_0)\otimes S(\pi_)(\delta\otimes\delta) \\
&= s_2(\delta\otimes\delta)
\end{aligned}$$

■

This not only provides an alternative way to describe a commutative coalgebra modality but also leads us into the definition of a monoidal storage category:

3.1.4. Definition. *A* **monoidal storage category** *is a symmetric monoidal category with a comonad* (S,δ,ϵ) *which has a storage transformation which is an isomorphism (*i.e. *both* s_2 *and* s_1 *are isomorphisms).*

Thinking of monoidal storage categories as "Seely categories", one notices that this is not the usual definition; in [M02] this definition (essentially) is given as a theorem which relates this description to the more standard definition given by Bierman and adapted by Schalk.

First, we shall compare monoidal storage categories to models of the multiplicative exponential fragment of linear logic (which we shall *not* assume includes being closed). We shall simply call these linear categories as they are Bierman's linear categories without the requirement of closure. Thus, a **linear category** is a symmetric monoidal category $\mathbb{X}$ with a coalgebra modality on a symmetric monoidal comonad $(S,\epsilon,\delta,m_\otimes,m_\top)$ such that Δ and e are both monoidal transformations and coalgebra morphisms.

The requirements of being a linear category are worth expanding. The requirement that $e\colon S(A)\to\top$ be a monoidal transformation amounts to two coherence diagrams:

$$\begin{tikzcd}
\top \arrow[r, "m_\top"] \arrow[dr, equal] & S(\top) \arrow[d, "e"] \\
& \top
\end{tikzcd}
\qquad
\begin{tikzcd}
S(A)\otimes S(B) \arrow[r, "m_\otimes"] \arrow[d, "e\otimes e"'] & S(A\otimes B) \arrow[d, "e"] \\
\top\otimes\top \arrow[r, "u"'] & \top
\end{tikzcd}$$

The requirement the $\Delta\colon S(A)\to S(A)\otimes S(A)$ is a monoidal transformation amounts to

two coherence requirements:

$$\begin{tikzcd}
\top \arrow[r, "m_\top"] \arrow[d, "u"'] & S(\top) \arrow[d, "\Delta"] \\
\top \otimes \top \arrow[r, "m_\top \otimes m_\top"'] & S(\top) \otimes S(\top)
\end{tikzcd}
\qquad
\begin{tikzcd}
S(A) \otimes S(B) \arrow[rr, "m_\otimes"] \arrow[d, "\Delta \otimes \Delta"'] & & S(A \otimes B) \arrow[dd, "\Delta"] \\
S(A) \otimes S(A) \otimes S(B) \otimes S(B) \arrow[d, "\mathrm{ex}_\otimes"'] & & \\
S(A) \otimes S(B) \otimes S(A) \otimes S(B) \arrow[rr, "m_\otimes \otimes m_\otimes"'] & & S(A \otimes B) \otimes S(A \otimes B)
\end{tikzcd}$$

Requiring that $e\colon S(A) \to \top$ forms a coalgebra morphism amounts to the requirement that

$$\begin{tikzcd}
S(A) \arrow[r, "\delta"] \arrow[d, "e"'] & S^2(A) \arrow[d, "S(e)"] \\
\top \arrow[r, "m_\top"'] & S(\top)
\end{tikzcd}$$

commutes. Finally requiring that Δ is a coalgebra morphism amounts to:

$$\begin{tikzcd}
S(A) \arrow[rrr, "\delta"] \arrow[d, "\Delta"'] & & & S^2(A) \arrow[d, "S(\Delta)"] \\
S(A) \otimes S(A) \arrow[r, "\delta \otimes \delta"'] & S^2(A) \otimes S^2(A) \arrow[r, "m_\otimes"'] & & S(S(A) \otimes S(A))
\end{tikzcd}$$

In his definition, Bierman had another requirement, namely that whenever $f\colon S(A) \to S(B)$ is a coalgebra morphism it must also be a comonoid morphism. We have amalgamated this into the definition of a coalgebra modality:

3.1.5. Lemma. *In a linear category there is the following implication of commutative diagrams*

$$\begin{tikzcd}
S(A) \arrow[r, "f"] \arrow[d, "\delta"'] & S(B) \arrow[d, "\delta"] \\
S^2(A) \arrow[r, "S(f)"'] & S^2(B)
\end{tikzcd}$$

$$\rule[0.5ex]{20em}{0.4pt}$$

$$\begin{tikzcd}
S(A) \arrow[r, "f"] \arrow[d, "\Delta"'] & S(B) \arrow[d, "\Delta"] \\
S(A) \otimes S(A) \arrow[r, "f \otimes f"'] & S(B) \otimes S(B)
\end{tikzcd}$$

Proof. We use the fact that δ is a morphism of comonoids and is a section (with retraction ϵ). This means the lower diagram commutes if and only if $f\Delta(\delta \otimes \delta) = \Delta(f \otimes f)(\delta \otimes \delta)$

but for this we have:

$$\begin{aligned} f\Delta(\delta\otimes\delta) &= f\delta\Delta = \delta S(f)\Delta = \delta\Delta(S(f)\otimes S(f)) \\ &= \Delta(\delta\otimes\delta)(S(f)\otimes S(f)) = \Delta((\delta S(f))\otimes(\delta(S(f)) = \Delta((f\delta)\otimes(f\delta)) \\ &= \Delta(f\otimes f)(\delta\otimes\delta). \end{aligned}$$

■

Note that this immediately means that if f is a coalgebra morphism, it is also then a comonoid morphism, and so Bierman's original definition of a linear category corresponds to ours (modulo the absence of closedness).

3.1.6. Theorem. *A monoidal storage category is a linear category and conversely a linear category with products is a monoidal storage category.*

Proof. We rely on a combination of [S04] and [M02] to provide the proof. In particular, the monoidal structure of the monad is given by.

$$\begin{aligned} \top \xrightarrow{m_\top} S(\top) &= \top \xrightarrow{s_1^{-1}} S(1) \xrightarrow{\delta} S^2(1) \xrightarrow{S(s_1)} S(\top) \\ S(A)\otimes S(B) \xrightarrow{m_\otimes} S(A\otimes B) &= S(A)\otimes S(B) \xrightarrow{s_2^{-1}} S(A\times B) \xrightarrow{\delta} S^2(A\times B) \\ &\xrightarrow{S(s_2)} S(S(A)\otimes S(B)) \xrightarrow{\epsilon\otimes\epsilon} S(A\otimes B) \end{aligned}$$

The converse requires checking that a linear category with products provides a storage isomorphism. The fact that there is a coalgebra modality is part of the data above. What remains is to prove that the induced storage transformation is an isomorphism and this is in the literature above. ■

3.2. Monoidal representation. We now return to the main thread of the paper and consider the notion of monoidal representation for a Cartesian storage category. Our objective is to show that a Cartesian storage category with monoidal representation is precisely the coKleisli category of a monoidal storage category. To achieve this, however, we must first take a detour to develop the notion of tensor representation. Recall that a basic intuition for a tensor product is that it should represent bilinear maps: clearly this intuition can be expressed in any category with a system of linear maps:

3.2.1. Definition.

(i) *In any Cartesian category a system of linear maps has* **monoidal representation** *in case for each X and Y there is an object $X\otimes Y$ and a bilinear map $\varphi_\otimes\colon X\times Y \to X\otimes Y$ such that for every bilinear map $g\colon X\times Y\to Z$ in $\mathbb{X}$ there is a unique linear map in $\mathbb{X}$ making the following diagram commute:*

$$\begin{tikzcd} X\times Y \arrow[r, "g"] \arrow[d, "\varphi_\otimes"'] & Z \\ X\otimes Y \arrow[ur, dotted, "g_{\otimes[A]}"'] & \end{tikzcd}$$

(ii) In any Cartesian category a system of linear maps has **strong monoidal representation** *in case for each X and Y there is an object $X \otimes Y$ and a bilinear map $\varphi_{\otimes}: X \times Y \longrightarrow X \otimes Y$ such that for every bilinear map $g: X \times Y \longrightarrow Z$ in $\mathbb{X}[A]$ there is a unique linear map in $\mathbb{X}[A]$ making the above diagram commute. Note that this means in $\mathbb{X}$ we have the diagram:*

$$\begin{tikzcd} A \times X \times Y \arrow[r, "g"] \arrow[d, "1\times\varphi_{\otimes}"'] & Z \\ A \times (X \otimes Y) \arrow[ur, dotted, "g_{\otimes[A]}"'] & \end{tikzcd}$$

(iii) A system of linear maps is **unit representable** *in case there is a linear map $\varphi_{\top}: 1 \longrightarrow \top$ such that in $\mathbb{X}$ for each point $p: 1 \longrightarrow Z$ there is a unique linear point $p_{\top}: \top \longrightarrow Z$ making*

$$\begin{tikzcd} 1 \arrow[r, "p"] \arrow[d, "\varphi_{\top}"'] & Y \\ \top \arrow[ur, dotted, "p^{\top}"'] & \end{tikzcd}$$

commute.

(iv) A system of linear maps is **strongly unit representable** *in case it is representable in each simple slice. This means there is a unique $p^{\top[A]}$ making*

$$\begin{tikzcd} A \times 1 \arrow[r, "p"] \arrow[d, "1\times\varphi_{\top}"'] & Y \\ A \times \top \arrow[ur, dotted, "p^{\top[A]}"'] & \end{tikzcd}$$

commute.

(v) A strong monoidal representation is **persistent** *in case, in (ii) above, whenever $A = A_1 \times B \times A_2$ and the map g is linear in B, then $g^{\otimes[A]}$ is linear in B. Similarly, for a strong monoidal unit representation to be persistent requires, setting $A = A_1 \times B \times A_2$ in (iv), that whenever p is linear in B then $p^{\top[A]}$ is also linear in B.*

As before, the basic form of monoidal representation assumes only that it holds in the original category: to be a strong monoidal representation requires the property must also hold in every simple slice. Thus, as for classification, there is a progression of notions: monoidal representation, strong monoidal representation, and persistent strong monoidal representation, each of which demands more than the last. This also applies to the unit representation; however, as we shall shortly discover (see Lemma 3.2.4), every storage category already has a persistent strong unit representation. Thus, we shall often talk of *monoidal representation* when we mean both monoidal and unit representation.

We first observe that persistence produces a simultaneous universal property:

3.2.2. Lemma. *If $f\colon A \times X \times Y \times Z \longrightarrow W$ is linear in its last two arguments, then there are unique maps linear in their last arguments, $f_1\colon A \times X \otimes (Y \otimes Z) \longrightarrow W$ and $f_2\colon A \times (X \otimes Y) \otimes Z \longrightarrow W$ such that $(1 \times ((1 \times \varphi)\varphi))f_1 = f = (1 \times ((\varphi \times 1)\varphi))f_2$.*

This is useful in the proof of:

3.2.3. Proposition. *If $\mathbb{X}$ has a system of linear maps for which there is a persistent strong monoidal (and unit) representation, then $\otimes$ is a symmetric tensor product with unit $\top$ on the subcategory of linear maps, $\mathcal{L}[]$.*

Proof. When we have a persistent representation we can define the required isomorphisms for a tensor product:

1. The associativity isomorphism:

$$\begin{tikzcd}
A \times B \times C \arrow[r, "\varphi_\otimes \times 1"] \arrow[d, "1\times\varphi_\otimes"'] & (A \otimes B) \times C \arrow[r, "\varphi_\otimes"] & (A \otimes B) \otimes C \\
A \times (B \otimes C) \arrow[d, "\varphi_\otimes"'] \arrow[urr, dotted] & & \\
A \otimes (B \otimes C) \arrow[uurr, dotted, "a_\otimes"'] & &
\end{tikzcd}$$

where the key is to observe that $(1 \times \varphi_\otimes)\varphi_\otimes\colon A \times B \times C \longrightarrow (A \otimes B) \otimes C$ is trilinear so that the two extensions indicated are given by the universal property.

2. The symmetry isomorphism:

$$\begin{tikzcd}
A \times B \arrow[r, "c_\times"] \arrow[d, "\varphi_\otimes"'] & B \times A \arrow[r, "\varphi_\otimes"] & B \otimes A \\
A \otimes B \arrow[urr, dotted, "c_\otimes"'] & &
\end{tikzcd}$$

3. The unit isomorphisms:

$$\begin{tikzcd}
A \times 1 \arrow[r, "\pi_1"] \arrow[d, "1\times\varphi_\top"'] & A \\
A \times \top \arrow[d, "\varphi_\otimes"'] \arrow[ur, dotted, "(\pi_0)^{[A]}_\top = \pi_0"'] & \\
A \otimes \top \arrow[uur, dotted, "u^R_\otimes"'] &
\end{tikzcd}$$

Note that $(\pi_0)^{[A]}_\top = \pi_0$ since both fit here and this is bilinear whence one can define $u^R_\otimes$. It is not obvious that $u^R_\otimes$ is an isomorphism, however, the diagram itself suggests that its inverse is

$$(u^R_\otimes)^{-1} = \pi_1^{-1}(1 \times \varphi_\top)\varphi_\otimes\colon A \otimes \top \longrightarrow A$$

and it is easily checked that this works.

We can now obtain the unit elimination on the left using the symmetry map $u^L_\otimes = c_\otimes u^R_\otimes$.

The coherences now follow directly from the fact that the product is a symmetric tensor and the multi-universal property of the representation. ∎

When a Cartesian storage category has strong monoidal representation, then this representation is automatically persistent. To establish this we start by observing that a Cartesian storage category already has a fair amount of monoidal representation.

3.2.4. Lemma. *In any Cartesian storage category*

(i) $\varphi\colon 1 \longrightarrow S(1)$ *gives persistent unit monoidal representation;*

(ii) $m_\times\colon S(A) \times S(B) \longrightarrow S(A \times B)$ *gives persistent monoidal representation for the objects* $S(A)$ *and* $S(B)$.

Proof. We shall focus on the binary monoidal representation: first note that $m_\times$ is bilinear as $m_\times = \theta S(\theta')\epsilon = \theta' S(\theta)\epsilon$ where the last two maps of each expansion are linear while θ is linear in its second argument while θ' is linear in its first.

The universal property for an arbitrary bilinear h is given by the following diagram, valid in any slice $\mathbb{X}[X]$:

$$\begin{tikzcd}
A \times B \arrow[r, "\varphi\times\varphi"] \arrow[dr, "\varphi"'] & S(A) \times S(B) \arrow[r, "h=m((\varphi\times\varphi)h)^\sharp"] \arrow[d, "m_\times"] & Z \\
& S(A \times B) \arrow[ur, dotted, "((\varphi\times\varphi)h)^\sharp"'] &
\end{tikzcd}$$

where h must be the solution to the simultaneous classification. But this makes the righthand triangle commute by uniqueness of the universal property.

Furthermore this is a persistent representation as the classification is persistent. ∎

Notice that we have shown that the Seely isomorphisms $s_0\colon S(1) \longrightarrow \top$ and $s_2\colon S(A\times B) \longrightarrow S(A) \otimes S(B)$ are present. These isomorphisms, which are so central to the structure of linear logic, play an important role in what follows.

The observation above means that a storage category always has (persistent and strong) representation of the tensor unit. Thus, we now focus on understanding monoidal representation. A key observation is:

3.2.5. Lemma. *In a Cartesian storage category* $\mathbb{X}$, $f\colon A\times B \longrightarrow C$ *is bilinear if and only if its classifying linear map* $f^\sharp\colon S(A \times B) \longrightarrow C$ *coequalizes the linear maps* $(S(\epsilon) \otimes S(\epsilon))s_2^{-1}$ *and* $(\epsilon \otimes \epsilon)s_2^{-1}$.

We shall use tensor notation even though we are not assuming that we have monoidal representation: this is justified by the fact that, for these particular objects, we are *always*

guaranteed representation by Lemma 3.2.4. The translation of these maps, $S(\epsilon) \otimes S(\epsilon)$ and $\epsilon_{S(A)} \otimes \epsilon_{S(A)}$, back into tensor-free notation uses the commuting diagrams below:

$$\begin{tikzcd}
S(S(A) \times S(B)) \arrow[r, "S(\epsilon\times\epsilon)"] \arrow[d, "s_2"'] & S(A \times B) \arrow[d, "s_2"] \\
S^2(A) \otimes S^2(B) \arrow[r, "S(\epsilon)\otimes S(\epsilon)"'] & S(A) \otimes S(B)
\end{tikzcd}$$

$$\begin{tikzcd}
S(S(A) \times S(B)) \arrow[r, "\epsilon"] \arrow[d, "s_2"'] & S(A) \times S(B) \arrow[r, "m_\times"] \arrow[dr, "\varphi_\otimes"'] & S(A \times B) \arrow[d, "s_2"] \\
S^2(A) \otimes S^2(B) \arrow[rr, "\epsilon\otimes\epsilon"'] & & S(A) \otimes S(B)
\end{tikzcd}$$

Proof. We shall use the characterization of bilinear maps in Proposition 2.3.7 to show that coequalizing these maps is equivalent to bilinearity.

Assume that $(S(\epsilon) \otimes S(\epsilon))s_2^{-1}f^\sharp = (\epsilon \otimes \epsilon)s_2^{-1}f^\sharp$ so that

$$(\varphi \times \varphi)\varphi_\otimes(S(\epsilon) \otimes S(\epsilon))s_2^{-1}f^\sharp = (\varphi \times \varphi)\varphi_\otimes(\epsilon \otimes \epsilon)s_2^{-1}f^\sharp$$

But then we have:

$$\begin{aligned}
&(\varphi \times \varphi)\varphi_\otimes(S(\epsilon) \otimes S(\epsilon))s_2^{-1}f^\sharp \\
&\quad = \quad (\varphi \times \varphi)(S(\epsilon) \times S(\epsilon))\varphi_\otimes s_2^{-1}f^\sharp \\
&\quad = \quad (\epsilon \times \epsilon)(\varphi \times \varphi)m_\times f^\sharp \\
&\quad = \quad (\epsilon \times \epsilon)\varphi f^\sharp \\
&\quad = \quad (\epsilon \times \epsilon)f \\
&(\varphi \times \varphi)\varphi_\otimes(\epsilon \otimes \epsilon)s_2^{-1}f^\sharp \\
&\quad = \quad (\varphi \times \varphi)(\epsilon \times \epsilon)\varphi_\otimes s_2^{-1}f^\sharp \\
&\quad = \quad \varphi_\otimes s_2^{-1}f^\sharp \\
&\quad = \quad m_\times S(f)\epsilon
\end{aligned}$$

So this condition implies bilinearity. Conversely, by reversing the argument, assuming bilinearity gives—using classification and monoidal representation—the equality of these maps. ∎

We now have:

3.2.6. Proposition. *$\mathbb{X}$ has (basic) monoidal representation at A and B if and only if*

$$\begin{tikzcd}
S^2(A) \otimes S^2(B) \arrow[r, shift left, "S(\epsilon) \otimes S(\epsilon)"] \arrow[r, shift right, "\epsilon \otimes \epsilon"'] & S(A) \otimes S(B) \arrow[r, "\epsilon \otimes \epsilon"] & A \otimes B
\end{tikzcd}$$

is a coequalizer in the subcategory of linear maps.

Proof. Suppose z equalizes $S(\epsilon) \otimes S(\epsilon)$ and $\epsilon \otimes \epsilon$ then by the previous lemma $z' = (\varphi \times \varphi)\varphi_{\otimes} z$ is bilinear and this determines a unique linear map $z'_{\otimes}$ with $z' = \varphi_{\otimes} z'_{\otimes}$.

$$\begin{tikzcd}
S^2(A) \times S^2(B) \arrow[r, shift left, "S(\epsilon)\times S(\epsilon)"] \arrow[r, shift right, "\epsilon\times\epsilon"'] \arrow[d, "\varphi_{\otimes}"'] & S(A) \times S(B) \arrow[r, "\epsilon\times\epsilon"] \arrow[d, "\varphi_{\otimes}"] & A \times B \arrow[d, "\varphi_{\otimes}"] \arrow[ddr, dotted, "z'"] & \\
S^2(A) \otimes S^2(B) \arrow[r, shift left, "S(\epsilon)\otimes S(\epsilon)"] \arrow[r, shift right, "\epsilon\otimes\epsilon"'] & S(A) \otimes S(B) \arrow[r, "\epsilon\otimes\epsilon"'] \arrow[drr, bend right, "z"'] & A \otimes B \arrow[dr, dotted, "z'_{\otimes}"] & \\
& & & Z
\end{tikzcd}$$

We claim $z'_{\otimes}$ is the unique comparison map making the fork a coequalizer in the linear map category. To show this we need $(\epsilon \otimes \epsilon) z'_{\otimes} = z$ and we must show that any other linear map k with $(\epsilon \otimes \epsilon)k = z$ has $\varphi_{\otimes} k = z'$. For the first of these we note that z is determined by representing $(\varphi \times \varphi)\varphi_{\otimes} z = z'$ but

$$(\varphi \times \varphi)\varphi_{\otimes}(\epsilon \otimes \epsilon) z'_{\otimes} = (\varphi \times \varphi)(\epsilon \times \epsilon)\varphi_{\otimes} z'_{\otimes} = \varphi_{\otimes} z'_{o} x = z'$$

so the two maps are equal. For the second, suppose we have such a k then

$$\varphi_{\otimes} k = (\varphi \times \varphi)(\epsilon \times \epsilon)\varphi_{\otimes} k = (\varphi \times \varphi)\varphi_{\otimes}(\epsilon \otimes \epsilon) k = (\varphi \times \varphi)\varphi_{\otimes} z = z'.$$

For the converse assume that the linear map category has this fork a coequalizer and suppose $f\colon A \times B \to C$ is bilinear. Returning to the diagram above, by Lemma 3.2.5 we may set $z' = f$ and $z = s_2^{-1} f^{\sharp}$. Noting that the top fork is an absolute coequalizer gives the map $\varphi_{\otimes}\colon A \times B \to A \otimes B$ which will then represent bilinear maps. ∎

Considering what happens in an arbitrary simple slice gives:

3.2.7. Corollary. *A storage category has strong monoidal representation if and only if forks of the form:*

$$X \times S^2(A) \otimes S^2(B) \underset{1 \times \epsilon \otimes \epsilon}{\overset{1 \times S(\epsilon) \otimes S(\epsilon)}{\rightrightarrows}} X \times S(A) \otimes S(B) \xrightarrow{1 \times \epsilon \otimes \epsilon} X \times A \otimes B$$

are coequalizers for the linear subcategories $\mathcal{L}[X]$

There are a number of different reasons why these forks might be coequalizers. It is often the case that these forks will also be coequalizers in the whole storage category. When this is so, one may decompose the presence of these coequalizers into the presence of the basic coequalizer (of Proposition 3.2.6) and the fact they must be preserved by the functor $X \times _$. Because these coequalizers are clearly reflexive, this latter condition is delivered whenever the product functor preserves, more generally, reflexive coequalization. Of course, when the storage category is Cartesian closed, the product functor will preserve *all* colimits and so in particular these coequalizers. In fact, the case which will be of

primary interest to us here, as shall be discussed below, is when these coequalizers are absolute and so are automatically preserved by the product functors and, furthermore, are present under the very mild requirement that linear idempotents split.

We have established that these coequalizers must be present in any storage category with strong monoidal representation. The final ingredient we need is persistence: fortunately this is guaranteed once strong monoidal representation is assumed.

3.2.8. Proposition. *In a Cartesian storage category with a strong monoidal representation, the representation is necessarily persistent: that is, given a map* $h\colon X_0\times C\times X_1\times A\times B \to Y$*, which is linear in* C*,* A*, and* B

$$\begin{tikzcd} X_0\times C\times X_1\times A\times B \arrow[r, "h"] \arrow[d, "1\times\varphi_\otimes"'] & Y \\ X_0\times C\times X_1\times (A\otimes B) \arrow[ur, "h^\otimes"'] & \end{tikzcd}$$

then its linear lifting $h_\otimes\colon X_0\times C\times X_1\times (A\otimes B)\to Y$ *is linear in* C*.*

Proof. By using Lemma 2.3.8 we may simplify what needs to be proven: namely, given $f\colon A\times B\times C\to Y$ which is linear in all its arguments then $f^\otimes\colon A\otimes B\times C\to Y$ is linear in C. This latter requires that we show that

$$\begin{tikzcd} A\otimes B\times S(C) \arrow[r, "\varphi\times 1"] \arrow[d, "1\times\epsilon"'] & S(A\otimes B)\times S(C) \arrow[r, "m_\times"] & S(A\otimes B\times C) \arrow[r, "f^\otimes"] & S(Y) \arrow[d, "\epsilon"] \\ A\otimes B\times C \arrow[rrr, "f^\otimes"'] & & & Y \end{tikzcd}$$

commutes. Because we have strong monoidal representation we know $(\epsilon\otimes\epsilon)\times 1\colon S(A)\otimes S(B)\times C \to A\otimes B\times C$ is epic. Thus, we may preface this square with the map $(s^2\times 1)((\epsilon\otimes\epsilon)\times 1)$ to test commutativity. Preliminary to this calculation note that whenever f is bilinear the following diagram

$$\begin{tikzcd} S(A)\times S(B) \arrow[r, "m_\times"'] \arrow[dr, "\varphi_\otimes"'] \arrow[dd, "\epsilon\times\epsilon"] & S(A\times B) \arrow[r, "S(f)"'] \arrow[d, "s_2"] & S(Y) \arrow[dd, "\epsilon"] \\ & S(A)\otimes S(B) \arrow[d, "\epsilon\otimes\epsilon"'] \arrow[dr, dotted] & \\ A\times B \arrow[r, "\varphi_\otimes"] \arrow[rr, bend right, "f"'] & A\otimes B \arrow[r, "f^\otimes"] & Y \end{tikzcd}$$

commutes as the dotted arrow is the unique linear extension of the bilinear map $m_\times S(f)\epsilon = (\epsilon\times\epsilon)f$. Putting the right square in the simple slice over C gives the following commuting

square:

$$\begin{tikzcd}
S(A \times B) \times C \arrow[r, "(1\times\varphi)m_\times"] \arrow[d, "s_2\times 1"'] & S(A \times B \times C) \arrow[r, "S(f)"] & S(Y) \arrow[dd, "\epsilon"] \\
S(A) \otimes S(B) \times C \arrow[d, "\epsilon\otimes\epsilon\times 1"'] & & \\
A \otimes B \times C \arrow[rr, "f_\otimes"'] & & Y
\end{tikzcd}$$

We now have the calculation:

$$\begin{aligned}
(s_2(\epsilon \otimes \epsilon) \times 1)(1 \times \epsilon)f^\otimes &= (1 \times \epsilon)(s_2(\epsilon \otimes \epsilon) \times 1)f^\otimes \\
&= (1 \times \epsilon)(1 \times \varphi)m_\times S(f)\epsilon \\
&= m_\times S(f)\epsilon \\
(s_2(\epsilon \otimes \epsilon) \times 1)(\varphi \times 1)m_\times S(f^\otimes)\epsilon &= (\varphi \times 1)m_\times S((s_2(\epsilon \otimes \epsilon) \times 1)f^\otimes)\epsilon \\
&= (\varphi \times 1)m_\times S((1 \times \varphi)m_\times S(f)\epsilon)\epsilon \\
&= (\varphi \times 1)m_\times S((1 \times \varphi)m_\times)\epsilon S(f)\epsilon \\
&= m_\times S(f)\epsilon
\end{aligned}$$

where the last step crucially uses the commutativity of the monad. ∎

When one has monoidal representation, if the tensor preserves the coequalizers which witness the exactness of the modality S, that is coequalizers of the form

$$S^2(A) \underset{\epsilon}{\overset{S(\epsilon)}{\rightrightarrows}} S(A) \xrightarrow{\epsilon} A$$

then the coequalizer above can be further analyzed *via* the following parallel coequalizer diagram:

$$\begin{tikzcd}
S^2(A) \otimes S^2(B) \arrow[r, shift left, "\epsilon\otimes 1"] \arrow[r, shift right, "S(\epsilon)\otimes 1"'] \arrow[d, shift right, "1\otimes S(\epsilon)"'] \arrow[d, shift left, "1\otimes\epsilon"] \arrow[dr, shift left, "\epsilon\otimes\epsilon"] \arrow[dr, shift right, "S(\epsilon)\otimes S(\epsilon)"'] & S(A) \otimes S^2(B) \arrow[r, "\epsilon\otimes 1"] \arrow[d, shift right, "1\otimes S(\epsilon)"'] \arrow[d, shift left, "1\otimes\epsilon"] & A \otimes S^2(B) \arrow[d, shift right, "1\otimes S(\epsilon)"'] \arrow[d, shift left, "1\otimes\epsilon"] \\
S^2(A) \otimes S(B) \arrow[r, shift left, "\epsilon\otimes 1"] \arrow[r, shift right, "S(\epsilon)\otimes 1"'] \arrow[u, bend left, "1\otimes\delta"] \arrow[d, "1\otimes\epsilon"] & S(A) \otimes S(B) \arrow[r, "\epsilon\otimes 1"] \arrow[d, "1\otimes\epsilon"] \arrow[dr, "\epsilon\otimes\epsilon"] & A \otimes S(B) \arrow[d, "1\otimes\epsilon"] \\
S^2(A) \otimes B \arrow[r, shift left, "\epsilon\otimes 1"] \arrow[r, shift right, "S(\epsilon)\otimes 1"'] & S(A) \otimes B \arrow[r, "\epsilon\otimes 1"'] & A \otimes B
\end{tikzcd}$$
$$\delta\otimes 1 : S^2(A) \otimes S^2(B) \to S^2(A) \otimes S(B)$$

As proven in [J02, Lemma 1.2.11], for example, given horizontal and vertical reflexive coequalizers as shown above, the diagonal is also a coequalizer. Recall that the basic coequalizers are actually absolute coequalizers in the storage category so they certainly are coequalizers in the linear category (use **LS.2**]) but they will not necessarily be absolute in the linear category. Thus, assuming that the tensor product preserves them is a far from benign assumption.

3.3. Codereliction and monoidal representation. An important source of monoidal representation in Cartesian storage categories arises from the presence of a **codereliction**: this is an unnatural transformation $\eta\colon A \longrightarrow S(A)$ which *is* natural for linear maps such that each η_A is linear and splits ϵ in the sense that $\eta_A\epsilon_A = 1_A$. Notice that φ, in general, will not be a codereliction: is natural for all maps but it is not in general linear—in fact if it were linear then all maps would be linear and S would necessarily be equivalent to the identity functor.

As we shall shortly see, all Cartesian differential storage categories have a codereliction. This is important because of the following observation:

3.3.1. Proposition. *Any Cartesian storage category in which linear idempotents split linearly, and which has a codereliction, has persistent monoidal representation.*

Recall that a linear idempotent e splits linearly when there is a splitting (r, s) with $rs = e$ and $sr = 1$ such that both s and r are linear. It follows from Lemma 2.5.1, that both r and s are linear if either is. Furthermore, it follows from Proposition 2.5.2 that we may always formally split linear idempotents.

Proof. Once we have a codereliction we have a split pair

$$\begin{tikzcd} S^2(A) \otimes S^2(B) \arrow[r, shift left, "\epsilon\otimes\epsilon"] \arrow[r, shift right, "S(\epsilon)\otimes S(\epsilon)"'] & S(A) \otimes S(B) \arrow[l, bend right, "\eta\otimes\eta"'] \end{tikzcd}$$

whose absolute coequalizer is the splitting of $\epsilon\eta \otimes \epsilon\eta$. Thus, when linear idempotents split one has basic monoidal representation. As this coequalizer is absolute in the whole storage category it is necessarily preserved by products and so the monoidal representation is strong, hence persistent. ∎

In particular, as we shall discover in Section 4, a Cartesian differential storage category always has a codereliction $\eta = \langle 1, 0\rangle D_\times[\varphi]$ which is linear (in the differential sense) and splits ϵ. Thus, in these examples it suffices for linear idempotents (in simple slices) to have linear splittings. This can always be formally arranged by splitting the linear idempotents.

3.4. Monoidal storage categories with an exact modality. We now observe that in a Cartesian storage category, which has monoidal representation, the linear maps form a monoidal storage category:

3.4.1. Proposition. *In any Cartesian storage category with monoidal representation, the subcategory of linear maps is a monoidal storage category.*

Proof. We have already observed that there is a monoidal comonad present on the linear maps. In addition, when both a persistent classification and representation is present there is are natural isomorphisms $s_{\otimes}\colon S(X \times Y) \to S(X) \otimes S(Y)$ and an isomorphism $s_{\top}\colon \top \to S(1)$ constituting an iso-comonoidal structure for the functor $S\colon \mathbb{X} \to \mathbb{X}$ from $\mathbb{X}$ with product to $\mathbb{X}$ with tensor: this is the storage isomorphism.

This follows immediately from the fact that

$$\begin{tikzcd} X \times Y \arrow[r, "\varphi\times\varphi"] \arrow[drr, "\varphi"'] & S(X) \times S(Y) \arrow[r, "\varphi_{\otimes}"] & S(X) \otimes S(Y) \arrow[d, dotted, "s_{\otimes}^{-1}"] \\ & & S(X \times Y) \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} X \times Y \arrow[r, "\varphi"] \arrow[dr, "(\varphi\times\varphi)\varphi_{\otimes}"'] & S(X \times Y) \arrow[d, dotted, "s_{\otimes}"] \\ & S(X) \otimes S(Y) \end{tikzcd}$$

have the same universal property. Similarly $\top$ and $S(1)$ have the same universal property. That this constitutes a storage transformation is straightforward to check. ∎

To establish the converse of this observation we need to prove that the coalgebra modality of a monoidal storage category is a comonad with a commutative force:

3.4.2. Proposition. *The coKleisli category of a monoidal storage category is a Cartesian storage category with monoidal representation.*

Proof. We shall show that the comonad is forceful where the force is defined as:

$$\begin{aligned} & S(A \times S(X)) \xrightarrow{\psi} S(A \times X) \\ = \;\; & S(A \times S(X)) \xrightarrow{s_2} S(A) \otimes S^2(X) \xrightarrow{1 \otimes \epsilon} S(A) \otimes S(X) \xrightarrow{s_2^{-1}} S(A \times X). \end{aligned}$$

We now have to check the six coherence diagrams of a force:

[Force.1] For the associativity of force we use the fact that the Seely isomorphism is comonoidal:

$$\begin{aligned} & \delta S(\sigma_{\times}) S(\epsilon \times \psi) S(\psi) S(a_{\times}) \\ = \;\; & \delta S(\sigma_{\times}) S(\epsilon \times 1) S(1 \times (s_2(1 \otimes \epsilon) s_2^{-1})) s_2 (1 \otimes \epsilon) s_2^{-1} S(a_{\times}) \\ = \;\; & \delta S(\sigma_{\times}) s_2 (\epsilon \otimes S(s_2(1 \otimes \epsilon) s_2^{-1}))(1 \otimes \epsilon) s_2^{-1} S(a_{\times}) \\ = \;\; & \delta S(\sigma_{\times}) s_2 (\epsilon \otimes (\epsilon s_2 (1 \otimes \epsilon)))(1 \otimes s_2^{-1}) s_2^{-1} S(a_{\times}) \\ = \;\; & \delta S(\sigma_{\times}) s_2 (\epsilon \otimes (\epsilon s_2 (1 \otimes \epsilon))) a_{\otimes} (s_2^{-1} \otimes 1) s_2^{-1} \\ = \;\; & \delta S(\sigma_{\times}) s_2 (\epsilon \otimes (\epsilon s_2))) a_{\otimes} (s_2^{-1} \otimes \epsilon) s_2^{-1} \\ = \;\; & s_2 (\delta \otimes \delta)(\epsilon \otimes (\epsilon s_2)) a_{\otimes} (s_2^{-1} \otimes \epsilon) s_2^{-1} \\ = \;\; & s_2 (1 \otimes s_2) a_{\otimes} (s_2^{-1} \otimes \epsilon) s_2^{-1} = S(a_{\times}) s_2 (1 \otimes \epsilon) s_2^{-1} = S(a_{\times}) \psi \end{aligned}$$

[Force.2] For projection of force:

$$\begin{aligned} \psi S(\pi_1) &= s_2 (1 \otimes \epsilon) s_2^{-1} S(\pi_1) \\ &= s_2 (1 \otimes \epsilon)(e \otimes 1) u_L^{\top} \\ &= s_2 (e \times 1) u_L^{\top} \epsilon \\ &= S(\pi_1) \epsilon \end{aligned}$$

[Force.3] For forceful naturality:

$$\begin{aligned}
\delta S(\sigma_\times)S(1\times(\epsilon\delta)))\psi &= \delta S(\sigma_\times)S(1\times(\epsilon\delta))s_2(1\otimes\epsilon)s_2^{-1}\\
&= \delta S(\sigma_\times)s_2(1\otimes S(\epsilon\delta))(1\otimes\epsilon)s_2^{-1}\\
&= s_2(\delta\otimes\delta)(1\otimes S(\epsilon\delta))(1\otimes\epsilon)s_2^{-1}\\
&= s_2(1\otimes\epsilon)(\delta\otimes\delta)s_2^{-1}\\
&= s_2(1\otimes\epsilon)s_2^{-1}\delta S(\sigma_\times)\\
&= \psi\delta S(\sigma_\times)
\end{aligned}$$

[Force.4] For forcefulness of counit:

$$\begin{aligned}
\psi\psi &= s_@(1\otimes\epsilon)s_2^{-1}s_2(1\otimes\epsilon)s_2^{-1}\\
&= s_2(1\otimes\epsilon\epsilon)s_2^{-1}\\
&= s_2(1\otimes S(\epsilon)\epsilon)s_2^{-1}\\
&= S(1\times\epsilon)s_2(1\otimes\epsilon)s_2^{-1}\\
&= S(1\times\epsilon)\psi
\end{aligned}$$

[Force.5] For forcefulness of comultiplication:

$$\begin{aligned}
\delta S(\sigma_\times)\psi S(\epsilon\times 1) &= \delta S(\sigma_\times)s_2(1\otimes\epsilon)s_2^{-1}S(\epsilon\times 1)\\
&= s_2(\delta\otimes\delta)(S(\epsilon)\otimes\epsilon)s_2^{-1} = 1
\end{aligned}$$

[Force.6] Commutativity of force:

$$\begin{aligned}
\psi\psi' &= s_2(1\otimes\epsilon)s_2^{-1}s^2(\epsilon\otimes 1)s_2^{-1}\\
&= s_2(1\otimes\epsilon)(\epsilon\otimes 1)s_2^{-1}\\
&= s_2(\epsilon\otimes 1)(1\otimes\epsilon)s_2^{-1}\\
&= s_2(\epsilon\otimes 1)s_2^{-1}s^2((1\otimes\epsilon)s_2^{-1}\\
&= \psi'\psi
\end{aligned}$$

This shows that the coKleisli category is a Cartesian storage category; it remains to show that it has monoidal representation. However, this follows, since in the coKleisli category $A\times B \xrightarrow{\varphi_\otimes} A\otimes B$ is given by the $\mathbb{X}$-map

$$S(A\times B) \xrightarrow{s_2} S(A)\otimes S(B) \xrightarrow{\epsilon\otimes\epsilon} A\otimes B$$

That this represents the tensor is easily seen, and persistence is automatic in a coKleisli category. ∎

The results above tell us that a monoidal storage category with an exact modality is always the category of linear maps of some Cartesian storage category with monoidal representation. This is because its coKleisli category is a Cartesian storage category with monoidal representation and one can recover, as the linear maps, the original monoidal storage category.

3.5. Closed storage categories. We briefly return to the issue of the closeness of these various categories. Starting with a monoidal storage category (Seely category) it is well-known that if the category is closed in the sense that there is an adjunction

$$\frac{\dfrac{A \otimes X \to B}{}}{X \to A \multimap B}$$

then the coKleisli category is Cartesian closed with

$$A \Rightarrow B := S(A) \multimap B$$

as we have the following natural equivalences:

$$\frac{\dfrac{\dfrac{S(A \times X) \to B}{}}{S(A) \otimes S(X) \to B}}{S(X) \to S(A) \multimap B}$$

Thus, if the monoidal storage category is closed then the coKleisli category, that is the Cartesian storage category, must be Cartesian closed. We wish to work for a converse, namely, knowing that the Cartesian storage category is Cartesian closed, can we provide some natural conditions for the linear maps to form a monoidal closed category? The conditions that we shall consider, in fact, do not require monoidal representation and, in fact, make the linear maps into a closed category in the original sense of Kelly and Eilenberg [KE65].

Note that if a Cartesian storage category is Cartesian closed, then the functor S is enriched over itself, the enrichment $S_{AB}\colon A \Rightarrow B \to S(A) \Rightarrow S(B)$ being the exponential adjoint to the composite map $S(A) \times (A \Rightarrow B) \xrightarrow{\theta'} S(A \times (A \Rightarrow B)) \xrightarrow{S(\mathsf{ev})} S(B)$.

3.5.1. Definition.

(i) A Cartesian storage category has **linear hom-objects** *if the following equalizer $A \multimap B$ exists for each A and B:*

$$A \multimap B \xrightarrow{k_{AB}} A \Rightarrow B \underset{\epsilon \Rightarrow 1}{\overset{S_{AB}\,(1 \Rightarrow \epsilon)}{\rightrightarrows}} S(A) \Rightarrow B$$

with k_{AB} linear for each A and B.

(ii) A Cartesian storage category is **closed** *if it is Cartesian closed, the linear maps form a closed system (see Definition 2.2.2), and linear hom-objects exist. We call such a category a Cartesian closed storage category.*

Clearly we are intending that the equalizer $A \multimap B$ should provide the object of linear maps from A to B. That it has this property relies on the following:

3.5.2. Proposition. *In any Cartesian closed storage category:*

(i) *If $f\colon A \times X \longrightarrow B$ is linear in its first argument then there is a unique map $\tilde{f}\colon X \longrightarrow A \multimap B$ making*

$$\begin{tikzcd}
A \times X \arrow[rr, "f"] \arrow[d, dotted, "1\times\tilde{f}"] & & B \\
A \times A \multimap B \arrow[r, "1\times k_{AB}"'] & A \times A \Rightarrow B \arrow[ur, "\mathsf{ev}"'] &
\end{tikzcd}$$

commute.

(ii) *Any f which can be expressed as $f = (1 \times \tilde{f})(1 \times k_{AB})\mathsf{ev}$ must be linear in its first argument.*

(iii) *If f is bilinear then $\tilde{f}$ is linear.*

Proof.

(i) Clearly $\tilde{f}k_{AB}$ must be the unique map to the hom-object $A \Rightarrow B$, $\hat{f} = \tilde{f}k_{AB}\colon X \longrightarrow A \Rightarrow B$, and so, as k_{AB} is monic, $\tilde{f}$, must be unique, if it exists. The difficulty is to show that $\hat{f}$ factors through this equalizer.

Recall that as f is linear in its first argument we have

$$\begin{tikzcd}
S(A) \times X \arrow[r, "\theta'"] \arrow[d, "\epsilon\times 1"'] & S(A \times X) \arrow[r, "S(f)"] & S(B) \arrow[d, "\epsilon"] \\
A \times X \arrow[rr, "f"'] & & B
\end{tikzcd}$$

commutes (that is $(1 \times \epsilon)f = \theta' S(f)\epsilon$). But then we have the following two commuting diagrams displaying the curried map for these two maps:

$$\begin{tikzcd}
S(A) \times X \arrow[r, "\epsilon\times 1"] \arrow[d, "!\times\hat{f}"'] & A \times X \arrow[d, "1\times\hat{f}"] \arrow[drr, bend left, "f"] & & \\
S(A) \times A \Rightarrow B \arrow[r, "\epsilon\times 1"'] \arrow[d, "1\times(\epsilon\Rightarrow 1)"'] & A \times A \Rightarrow B \arrow[rr, "\mathsf{ev}"'] & & B \\
S(A) \times S(A) \Rightarrow B \arrow[urrr, bend right, "\mathsf{ev}"'] & & &
\end{tikzcd}$$

This shows that $\mathsf{curry}((1 \times \epsilon)f) = \hat{f}(\epsilon \Rightarrow 1)$.

$$\begin{tikzcd}
S(A) \times X \arrow[r, "\theta'"] \arrow[d, "1\times\hat{f}"'] & S(A \times X) \arrow[r, "S(f)"] & S(B) \arrow[dr, "\epsilon"] & \\
S(A) \times A \Rightarrow B \arrow[d, "1\times S_{AB}"'] & & & \\
S(A) \times S(A) \Rightarrow S(B) \arrow[d, "1\times(1\Rightarrow\epsilon)"'] \arrow[uurr, bend left, "\mathsf{ev}"] & & & B \\
S(A) \times S(A) \Rightarrow B \arrow[urrr, bend right, "\mathsf{ev}"'] & & &
\end{tikzcd}$$

While this shows that $\mathsf{curry}(\theta' S(f)\epsilon) = \hat{f} S_{AB}(1 \Rightarrow \epsilon)$. Thus $\hat{f}$ factors through the equalizer as desired.

(ii) For the converse we have:

$$\begin{tikzcd}
S(A) \times X \arrow[r, "\theta'"] \arrow[dr, "!\times\hat{f}"'] \arrow[ddd, "\epsilon\times 1"'] & S(A \times X) \arrow[rr, "S(f)"] & & S(B) \arrow[ddd, "\epsilon"] \\
& S(A) \times A \multimap B \arrow[r, "1\times(k_{AB}S_{AB})"] \arrow[dr, "1\times(k_{AB}(\epsilon\Rightarrow 1))"'] & S(A) \times S(A) \Rightarrow S(B) \arrow[ur, "\mathsf{ev}"] \arrow[d, "1\times(1\Rightarrow\epsilon)"] & \\
& & S(A) \times S(A) \Rightarrow B \arrow[dr, "\mathsf{ev}"] & \\
A \times X \arrow[rrr, "f"'] & & & B
\end{tikzcd}$$

(iii) It remains to show that $\tilde{f}$ is linear when f is bilinear that is $\epsilon\tilde{f} = S(\tilde{f})\epsilon$ for which we have:

$$\begin{tikzcd}
A \times S(X) \arrow[r, "1\times\epsilon"] \arrow[d, "1\times\epsilon\tilde{f}"'] & A \times X \arrow[d, "f"] \arrow[dl, "1\times\tilde{f}"'] \\
A \times A \multimap B \arrow[r, "\mathsf{ev}_{\multimap}"'] & B
\end{tikzcd}$$

$$\begin{tikzcd}
A \times S(X) \arrow[r, "\theta"] \arrow[d, "1\times S(\tilde{f})"'] & S(A \times X) \arrow[r, "S(f)"] & S(B) \arrow[ddd, "\epsilon"] \\
A \times S(A \multimap B) \arrow[r, "\theta"'] \arrow[d, "1\times\epsilon"'] & S(A \times A \multimap B) \arrow[ur, "S(\mathsf{ev}_{\multimap})"'] & \\
A \times A \multimap B \arrow[rr, "\mathsf{ev}_{\multimap}"'] & & B
\end{tikzcd}$$

Showing linearity in the second argument implies that $\tilde{f}$ is linear.

■

In particular, notice that this immediately means that $\mathsf{ev}_{\multimap} := (1 \times k_{AB})\mathsf{ev}$ is linear in its first argument, as $\mathsf{ev}_{\multimap}$ defined in this manner certainly satisfies Proposition 3.5.2 *(ii)*. However, as k_{AB} is linear and ev is linear in its second argument—as we are assuming a closed linear system—it follows that $\mathsf{ev}_{\multimap}$ is bilinear. Thus, when one has monoidal representation this means that there is an induced evaluation map $\mathsf{ev}_{\otimes}: A \otimes (A \multimap B) \longrightarrow B$ with $\varphi_{\otimes}\mathsf{ev}_{\otimes} = \mathsf{ev}_{\multimap}$ which is linear. Furthermore, by Proposition 3.5.2 (iii) the curry map for $f: A \otimes X \longrightarrow B$ is also linear as it is $\widetilde{\varphi_{\otimes} f}$. Thus we have:

3.5.3. Corollary. *If $\mathbb{X}$ is a Cartesian closed storage category which has monoidal representation, then the subcategory of linear maps forms a monoidal closed storage category.*

This leaves open the converse: if one starts with a monoidal closed storage category, will its coKleisli category be a Cartesian closed storage category? We shall now show that this is true, giving a complete characterization for the closed case:

3.5.4. Proposition. *If $\mathbb{X}$ is a monoidal storage category which is monoidal closed then its coKleisli category is a Cartesian closed storage category.*

Proof. We provide a sketch of the proof. We must show three things:

(1) We need to show that the coKleisli category is a Cartesian closed category and that the functor induced by the modality is suitably enriched. As above, the closed structure is given by $A \Rightarrow B := S(A) \multimap B$. In a Cartesian closed category a functor is enriched whenever it is strong, so the fact that there is a natural force $\psi: S(A \times S(B)) \longrightarrow S(A \times B)$ (described in Proposition 3.4.2) guarantees the functor is strong.

(2) We need to show that the linear maps of the coKleisli category form a closed system. We know already they form a linear system so we need only check that they are a *closed* linear system. This amounts to checking the following.

   (a) The evaluation map is linear in its higher-order argument. This is immediately true by inspection. Here is the definition of the evaluation map in the coKleisli category:

$$S(A \times (S(A) \multimap B)) \xrightarrow{s_2} S(A) \otimes S(S(A) \multimap B) \xrightarrow{1 \otimes \epsilon} S(A) \otimes (S(A) \multimap B) \xrightarrow{\mathsf{ev}_{\multimap}} B$$

   The second map in the sequence ensures it is linear in the second argument.

   (b) Linearity is "persistent" over currying. To say that a coKleisli map $f: S(A \times B \times C) \longrightarrow D$ is linear in its second argument amounts to saying that $f = s_3(1 \otimes \epsilon \otimes 1)f'$. Currying this map (with respect to the first argument) clearly maintains the linearity in the second argument.

(3) This leaves only the requirement that $A \multimap B$ occurs as an equalizer:

$$A \multimap B \xrightarrow{k_{AB}} A \Rightarrow B \underset{\epsilon \Rightarrow 1}{\overset{S_{AB}(1 \Rightarrow \epsilon)}{\rightrightarrows}} S(A) \Rightarrow B$$

Recall that we may assume the modality is exact (if it is not, simply work in the subcategory of linear maps). This equalizer (after considerable unwinding) is exactly the image of

$$A \multimap B \xrightarrow{\epsilon \multimap 1} S(A) \multimap B \underset{\epsilon \multimap 1}{\overset{S(\epsilon) \multimap 1}{\rightrightarrows}} S^2(A) \multimap B$$

under the inclusion into the coKleisli category. Since

$$S^2(A) \underset{\epsilon}{\overset{S(\epsilon)}{\rightrightarrows}} S(A) \xrightarrow{\epsilon} A$$

under this inclusion is a coequalizer, the transpose of this is an equalizer. This shows that $A \multimap B$ does occur as an equalizer as required.

■

4. Cartesian differential storage categories

Monoidal differential categories were introduced in [BCS06]: they consist of a symmetric monoidal category with a coalgebra modality and a deriving transformation. In [BCS09] the notion of a Cartesian differential category was introduced. It was proven that an important way in which Cartesian differential categories arise is as the coKleisli categories of monoidal differential categories which satisfied an additional interchange requirement on the deriving transformation. This latter requirement was presented as a sufficient condition: the question of whether it was necessary was left open. Here we partially answer that question for an important class of monoidal differential categories: those whose modality is a storage modality—in other words those which are "Seely categories" in the sense discussed above—and exact: we show that this condition is indeed necessary.

Not surprisingly, the strategy of the proof is to characterize the coKleisli category of a monoidal differential category with a storage modality using the development above. The fact that the coKleisli category of a monoidal differential category (satisfying the interchange condition) is necessarily a Cartesian differential category means we should consider Cartesian storage categories which are simultaneously Cartesian differential categories. Furthermore, we expect the linear maps in the differential sense to be the linear maps in the storage sense. We will then show that, under these assumptions, the linear maps form a monoidal differential category which satisfies the interchange requirement. This then provides a *characterization* of coKleisli categories of monoidal differential categories (with a storage modality) which form differential categories.

It is worth mentioning—not least as it caused the authors some strife—that there are at least three different sorts of monoidal differential category which have been discussed in the literature:

(A) A differential given by a **deriving transformation** as introduced in [BCS06].

(B) A differential given by a **codereliction** as introduced [ER05] and described in [BCS06]

(C) A differential given by a **creation operator** as introduced in [F07].

Each form of differential is more specialized than its predecessor: the first merely requires a coalgebra modality, the second requires a bialgebra modality, while the last requires a storage modality. A standard way in which a bialgebra modality arises is from the presence of biproducts together with a storage modality: the Seely isomorphism then transfers the bialgebra structure of the biproduct onto the modality. This means that the last two forms of differential are particularly suited to storage settings. As, in the current context, we are considering storage categories one might think that the differential which one extracts from a Cartesian differential storage category should be of type (C) or at least (B). It is, therefore, perhaps somewhat unexpected that the differential that emerges is one of type (A).

Cartesian differential storage categories are important in their own right. Their structure can be approached in two very different ways: from the perspective of being a Cartesian differential category with monoidal representation, or as the coKleisli category of a monoidal differential category. This tension allows one to transfer arguments between the Cartesian and the monoidal worlds. This ability is frequently used in differential geometry where one often wants to view differential forms as maps from tensor products.

4.1. Cartesian differential categories. Cartesian differential categories, were introduced in [BCS09]: here we briefly review their properties.

One way to view the axiomatization of Cartesian differential categories is as an abstraction of the Jacobian of a smooth map $f\colon \mathbb{R}^n \to \mathbb{R}^m$. One ordinarily thinks of the Jacobian as a smooth map

$$J(f)\colon \mathbb{R}^n \to \mathrm{Lin}(\mathbb{R}^n, \mathbb{R}^m).$$

Uncurrying, this means the Jacobian can also be seen as a smooth map

$$J(f)\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m$$

which is linear in its first variable. A Cartesian differential category asks for an operation of this type, satisfying the axioms described below. However, notice that to express the axioms, one needs the ability to add parallel maps. It turns out this is not an enrichment in commutative monoids as one might expect, but rather a skew enrichment, [Street], which makes it a Cartesian left additive category. In such a category the addition of arrows is only preserved by composition on the left, that is $f(g+h) = fg + gh$ and $f0 = 0$.

4.1.1. Definition. *A* ***Cartesian differential category*** *is a Cartesian left additive category with an operation*

$$\frac{X \xrightarrow{f} Y}{X \times X \xrightarrow[D[f]]{} Y}\ D[_]$$

(called "differentiation") satisfying:

[CD.1] $D[f+g] = D[f] + D[g]$ *and* $D[0] = 0$ *(differentiation preserves addition);*

[CD.2] $\langle a+b, c\rangle D[f] = \langle a, c\rangle D[f] + \langle b, c\rangle D[f]$ *and* $\langle 0, a\rangle D[f] = 0$ *(a derivative is additive in its first variable);*

[CD.3] $D[1] = \pi_0$, $D[\pi_0] = \pi_0\pi_0$, *and* $D[\pi_1] = \pi_0\pi_1$ *(projections and identity are linear);*

[CD.4] $D[\langle f, g\rangle] = \langle D[f], D[g]\rangle$ *(differentiation is compatible with pairing) ;*

[CD.5] $D[fg] = \langle D[f], \pi_1 f\rangle D[g]$ *(the chain rule);*

[CD.6] $\langle\langle a, 0\rangle, \langle c, d\rangle\rangle D[D[f]] = \langle a, d\rangle D[f]$ *(the differential is linear);*

[CD.7] $\langle\langle a, b\rangle, \langle c, d\rangle\rangle D[D[f]] = \langle\langle a, c\rangle, \langle b, d\rangle\rangle D[D[f]]$ *(interchange rule);*

We recall a number of examples of Cartesian differential categories:

4.1.2. Example.

(1) If $\mathbb{X}$ is an additive Cartesian category (*i.e.* enriched in commutative monoids) then, by defining $D[f] = \pi_0 f$, it can be viewed as a Cartesian differential category in which every map is linear.

(2) Smooth functions on finite dimensional Euclidean vector spaces form a Cartesian differential category.

(3) The coKleisli category of any monoidal differential category (satisfying the interchange law) is a Cartesian differential category, [BCS09]. A basic example of a monoidal differential category is provided by the category of relations, Rel, with respect to the "multi-set" (or "bag") comonad. The deriving transformation $d_{\otimes}\colon A \otimes M(A) \longrightarrow M(A)$, which adds another element to the multi–set $M(A)$, provides the (monoidal) differential structure.

(4) Convenient vector spaces [BET12] form a Cartesian differential category.

(5) There is a comonad Faa on the category of Cartesian left additive categories whose coalgebras are exactly Cartesian differential categories [CS11].

Cartesian differential categories already have a notion of "linear map": namely those f such that $D[f] = \pi_0 f$. Furthermore, it is a basic result that any simple slice, $\mathbb{X}[A]$, of a Cartesian differential category, $\mathbb{X}$, is also a Cartesian differential category. The differential in the simple slice $\mathbb{X}[A]$ is given by:

$$\frac{X \times A \xrightarrow{f} Y}{(X \times X) \times A \xrightarrow[D_A[f]]{} Y} \; D_A[_]$$

where

$$\begin{aligned} &(X \times X) \times A \xrightarrow{D_A[f]} Y \\ &= \quad (X \times X) \times A \xrightarrow[\langle\langle \pi_0\pi_0, 0\rangle, \langle \pi_0\pi_1, \pi_1\rangle\rangle]{} (X \times A) \times (X \times A) \xrightarrow[D[f]]{} Y \end{aligned}$$

This is precisely the familiar notion of a *partial derivative*.

In particular, this means we may isolate linear maps in this differential sense in each slice as those maps with $D_A[f] = (\pi_0 \times 1)f$. Our first observation is then:

4.1.3. Proposition. *In a Cartesian differential category, $\mathbb{X}$, the linear maps form a system of linear maps.*

Proof. We must check the requirements of being a linear system:

[LS.1] We require, for each $A \in \mathbb{X}$, that the identity maps, projections and pairings of linear maps are linear. As explained above, in $\mathbb{X}[A]$ a map $f\colon X \times A \to Y$ is linear in case $D_A[f] = (\pi_0 \times 1)f\colon (X \times X) \times A \to Y$ where $D_A[f]$ is the partial derivative of f:

$$D_A[f] := \langle\langle \pi_0\pi_0, 0\rangle, \langle \pi_0\pi_1, \pi_1\rangle\rangle D[f]$$

We shall leave all but the last requirement to the reader for which we shall do a concrete calculation in $\mathbb{X}$. Suppose f and g are linear in this sense in $\mathbb{X}[A]$ (that is they are in $\mathcal{L}[A]$), then we must show that the pairing of f and g in $\mathbb{X}[A]$ is linear: as a map in $\mathbb{X}$ this pairing is just $\langle f, g\rangle$ so that:

$$\begin{aligned} D_A[\langle f, g\rangle] &= \langle\langle \pi_0\pi_0, 0\rangle, \langle \pi_0\pi_1, \pi_1\rangle\rangle D[\langle f, g\rangle] \\ &= \langle\langle \pi_0\pi_0, 0\rangle, \langle \pi_0\pi_1, \pi_1\rangle\rangle \langle D[f], D[g]\rangle \\ &= \langle\langle\langle \pi_0\pi_0, 0\rangle, \langle \pi_0\pi_1, \pi_1\rangle\rangle D[f], \langle\langle \pi_0\pi_0, 0\rangle, \langle \pi_0\pi_1, \pi_1\rangle\rangle D[g]\rangle \\ &= \langle D_A[f], D_A[g]\rangle \\ &\qquad \text{(Using linearity of } f \text{ and } g\text{)} \\ &= \langle\langle (\pi_0 \times 1)f, (\pi_0 \times 1)g\rangle \\ &= (\pi_0 \times 1)\langle f, g\rangle \end{aligned}$$

showing that $\langle f, g\rangle$ is linear in this sense.

[LS.2] We must show that these linear maps are closed to composition in each slice and that when g is a linear retraction that $gh \in \mathcal{L}[A]$ implies $h \in \mathcal{L}[A]$. Leaving composition to the reader let us focus on the second part. If $sg = 1$ then using the non-trivial fact that $D_A[_]$ is a differential in $\mathbb{X}[A]$ we have

$$\begin{aligned} D_A[h] &= D_A[sgh] = \langle D_A[s], \pi_1 s\rangle D_A[gh] = \langle D_A[s], \pi_1 s\rangle \pi_0 gh \\ &= D_A[s]gh = D_A[sg]h = D_A[1]h = \pi_0 h. \end{aligned}$$

[LS.3] For the last part we shall do a concrete calculation in $\mathbb{X}$: we must show that if h is linear in $\mathbb{X}[B]$ then $\mathbb{X}[f](h)$ is linear in $\mathbb{X}[A]$; here is the calculation:

$$\begin{aligned} D_A[\mathbb{X}[f](h)] &= D_A[(1 \times f)h] = \langle\langle \pi_0\pi_0, 0\rangle, \langle \pi_0\pi_1, \pi_1\rangle\rangle D[(1 \times f)h] \\ &= \langle\langle \pi_0\pi_0, 0\rangle, \langle \pi_0\pi_1, \pi_1\rangle\rangle \langle D[1 \times f], \pi_1(1 \times f)\rangle D[h] \\ &= \langle\langle \pi_0\pi_0, 0\rangle, \langle \pi_0\pi_1, \pi_1\rangle\rangle \langle\langle \pi_0\pi_0, \langle \pi_0\pi_1, \pi_1\pi_1\rangle D[f]\rangle, \pi_1(1 \times f)\rangle D[h] \\ &= \langle\langle \pi_0\pi_0, 0\rangle, \langle \pi_0\pi_1, \pi_1\rangle\rangle \langle\langle \pi_0\pi_0, \langle \pi_0\pi_1, \pi_1\pi_1\rangle D[f]\rangle, \pi_1(1 \times f)\rangle D[h] \\ &= \langle\langle \pi_0\pi_0, \langle 0, \pi_1\rangle D[f]\rangle, \langle \pi_0\pi_1, \pi_1\rangle(1 \times f)\rangle D[h] \\ &= \langle\langle \pi_0\pi_0, 0\rangle, \langle \pi_0\pi_1, \pi_1\rangle(1 \times f)\rangle D[h] \\ &= (1 \times f)\langle\langle \pi_0\pi_0, 0\rangle, \langle \pi_0\pi_1, \pi_1\rangle\rangle D[h] \\ &\qquad \text{(Linearity of } h) \\ &= (1 \times f)(\pi_0 \times 1)h \\ &= (\pi_0 \times 1)(1 \times f)h \end{aligned}$$

■

4.2. Cartesian differential storage categories: the basics. A **Cartesian differential storage category** is a Cartesian differential category whose linear maps—in the natural differential sense—are (strongly and persistently) classified.

We first observe that every Cartesian differential category has a codereliction map defined by:

$$\eta_A := \langle 1, 0\rangle D[\varphi] : A \to S(A).$$

4.2.1. Lemma. *η so defined is a codereliction map: that is it is natural for linear maps and has $\eta\epsilon = 1$.*

Proof. We need to show $\eta_A \epsilon_A = 1_A$:

$$\begin{aligned} \eta_A \epsilon_A &= \langle 1, 0\rangle D[\varphi]\epsilon \\ &= \langle 1, 0\rangle D[\varphi]\epsilon] \qquad \text{(as } \epsilon \text{ is linear)} \\ &= \langle 1, 0\rangle D[1] = \langle 1, 0\rangle \pi_0 = 1 \end{aligned}$$

and that, if $f\colon A \to B$ is a linear map, that $f\eta_B = \eta_A S(f)$:

$$\begin{aligned}
\eta_A S(f) &= \langle 1,0\rangle D[\varphi]S(f) \\
&= \langle 1,0\rangle D[\varphi S(f)] \qquad \text{(as } S(f) \text{ is linear)} \\
&= \langle 1,0\rangle D[f\varphi] \\
&= \langle 1,0\rangle \langle D[f], \pi_1 f\rangle D[\varphi] \\
&= \langle 1,0\rangle \langle \pi_0 f, \pi_1 f\rangle D[\varphi] \\
&= \langle 1,0\rangle \langle \pi_0 f, \pi_1 f\rangle D[\varphi] \\
&= \langle f, 0f\rangle D[\varphi] = f\langle 1,0\rangle D[\varphi] = f\eta_B
\end{aligned}$$

■

Because Cartesian differential storage categories always have a codereliction map, up to splitting linear idempotents, they also have monoidal representation. Thus it makes sense to ask whether the subcategory of linear maps forms a monoidal differential category. To prove this is so will be the main aim of the current section. However, before turning to this it is worth making some further basic observations.

4.2.2. Lemma. *If $\mathbb{X}$ is a Cartesian differential storage category with monoidal representation then the tensor product on the linear maps is additive in the sense that $(f+g)\otimes h = f\otimes h + g\otimes h$ and $0\otimes h = 0$.*

Proof. We illustrate the proof with the first equation: $(f+g)\otimes h = (f\otimes h) + (g\otimes h)$. This follows from the fact that $(f+g)\otimes h$ and $(f\otimes h)+(g\otimes h)$ are determined respectively by$\varphi_\otimes((f+g)\otimes h)$ and $\varphi_\otimes((f\otimes h)+(g\otimes h))$. So:

$$\begin{aligned}
\varphi_\otimes((f+g)\otimes h) &=_{[\text{defn}]} ((f+g)\times h)\varphi_\otimes \\
&=_{[\varphi_\otimes \text{ bilinear}]} (f\times h)\varphi_\otimes + (g\times h)\varphi_\otimes \\
&=_{[\text{defn}]} \varphi_\otimes(f\otimes h) + \varphi_\otimes(g\otimes h) \\
&=_{[\text{left additive}]} \varphi_\otimes((f\otimes h)+(g\otimes h))
\end{aligned}$$

The 0-ary case is similar. ■

Suppose $\mathbb{X}$ is a Cartesian storage category which is also a Cartesian differential category. In this context, $\mathbb{X}$, has two notions of "linear" map: *viz* the notion which all storage categories have, and the notion that all Cartesian differential categories have. An interesting observation is:

4.2.3. Proposition. *$\mathbb{X}$ is a Cartesian differential storage category if and only if it is both a Cartesian differential and storage category and $S(f)$ (for any f) and ϵ are linear in the differential sense and the codereliction map η is linear in the storage sense.*

Proof. By assumption, $D[S(f)] = \pi_0 S(f)$, for any f, $D[\epsilon] = \pi_0\epsilon$, and η is ϵ-natural. We must show that f is ϵ-natural if and only if $D_\times[f] = \pi_0 f$. So suppose first that f is ϵ-natural:

$$\begin{aligned} D[f] &= (\varphi \times \varphi)(\epsilon \times \epsilon)D[f] \\ &= (\varphi \times \varphi)D[\epsilon f] \\ &= (\varphi \times \varphi)D[S(f)\epsilon] \\ &= (\varphi \times \varphi)\pi_0 S(f)\epsilon \\ &= (\varphi \times \varphi)\pi_0 \epsilon f \\ &= \pi_0 \varphi \epsilon f = \pi_0 f \end{aligned}$$

Next, suppose that $D[f] = \pi_0 f$ we must show f is ϵ-natural assuming that η is. We observe that, in this case, $f = \eta S(f)\epsilon$, from which the result is immediate, as η, $S(f)$, and ϵ are ϵ natural. Here is the calculation:

$$\begin{aligned} f &= \langle 1, 0\rangle \pi_0 f \\ &= \langle 1, 0\rangle D[f] \\ &= \langle 1, 0\rangle D[f\varphi\epsilon] \\ &= \langle 1, 0\rangle D[\varphi S(f)\epsilon] \\ &= \langle 1, 0\rangle D[\varphi] S(f)\epsilon \\ &= \eta S(f)\epsilon \end{aligned}$$

■

Recall that in a monoidal differential category it is possible to re-express the derivative using a *deriving transformation* $d_\otimes\colon A \otimes S(A) \to S(A)$ into a more compact form. It is natural to wonder whether the same thing cannot be done for Cartesian differential storage categories: that is define the derivative in terms of analogous structure:

4.2.4. Definition. *A* **Cartesian deriving transformation** *on a Cartesian storage category is a (not-necessarily natural) transformation* $d_\times\colon A \times A \to S(A)$ *satisfying:*

[cd.1] $d_\times S(0)\epsilon = 0$, $d_\times S(f+g)\epsilon = d_\times(S(f) + S(g))\epsilon$

[cd.2] $\langle h+k, v\rangle d_\times = \langle h, v\rangle d_\times + \langle k, v\rangle d_\times$, $\langle 0, v\rangle d_\times = 0$

[cd.3] $d_\times \epsilon = \pi_0$

[cd.4] $d_\times S(\langle f, g\rangle)\epsilon = d_\times\langle S(f)\epsilon, S(g)\epsilon\rangle$ *(Note that* $d_\times S(!)\epsilon = d_\times! =!$ *is true since* 1 *is terminal.)*

[cd.5] $d_\times S(fg)\epsilon = \langle d_\times S(f)\epsilon, \pi_1 f\rangle d_\times S(g)\epsilon$

[cd.6] $\langle\langle g, 0\rangle, \langle h, k\rangle\rangle d_\times S(d_\times)\epsilon = \langle g, k\rangle d_\times$

[cd.7] $\langle\langle 0,h\rangle,\langle g,k\rangle\rangle d_\times S(d_\times)\epsilon = \langle\langle 0,g\rangle,\langle h,k\rangle\rangle d_\times S(d_\times)\epsilon$

[cd.8] $\eta =_{\mathsf{def}} \langle 1,0\rangle d_\times$ *is ϵ-natural (or linear).*

It is now straightforward to observe:

4.2.5. Proposition. *A Cartesian storage category with a Cartesian deriving transformation is precisely a Cartesian differential storage category.*

Proof. The translation between the two structures is given by:

$$D[f]:= d_\times S(f)\epsilon \quad \text{and} \quad d_\times := D[\varphi]$$

Note that these are inverse as

$$D[f]:= d_\times S(f)\epsilon := D[\varphi]S(f)\epsilon = D[\varphi S(f)\epsilon] = D[f\varphi\epsilon] = D[f]$$

and

$$d_\times := D[\varphi]:= d_\times S(\varphi)\epsilon = d_\times.$$

Most of the axioms are clearly direct translations of each other: **[CD.1,2,4-7]** and **[cd.1,2,4-7]** are clearly equivalent through this translation. For **[CD.3]**, note that $D[1] = d_\times\epsilon = \pi_0$ and that $D[\pi_i] = d_\times S(p_i)\epsilon = d_\times\epsilon\pi_i = \pi_0\pi_i$, since in a storage category, projections are linear (by **[LS.1]**).

It remains to show that being linear in the differential sense coincides with being linear in the storage sense. First note if f is epsilon natural, that is $\epsilon f = S(f)\epsilon$, then $D[f] = d_\times S(f)\epsilon = d_\times\epsilon f = \pi_0 f$. Conversely, suppose that $D_\times[f] = \pi_0 f$, then $f = \eta S(f)\epsilon$ as

$$f \;=\; \langle 1,0\rangle\pi_0 f \;=\; \langle 1,0\rangle d_\times S(f)\epsilon \;=\; \eta S(f)\epsilon$$

which by assumption is linear (in the sense of being ϵ-natural). ∎

4.3. The main theorem. From the results of Section 3, we also note that, in the presence of a codereliction map and when sufficient linear idempotents split, every Cartesian differential storage category has strong persistent monoidal representation. The main result of the paper is:

4.3.1. Theorem. *The linear maps of a Cartesian differential storage category, in which linear idempotents split, form a monoidal differential storage category satisfying the interchange rule.*

Let us first recall what a monoidal differential category is. A **monoidal differential category** is a monoidal category with a coalgebra modality (see Definition 3.1.2) equipped with a natural transformation

$$d_\otimes\colon A\otimes S(A)\to S(A)$$

called a **deriving transformation** satisfying:

[d.1] $d_\otimes e = 0$ (constants)

[d.2] $d_\otimes \epsilon = 1 \otimes e$ (linear maps)

[d.3] $d_\otimes \Delta = (1 \otimes \Delta)(d_\otimes \otimes 1) + (1 \otimes \Delta)(c_\otimes \otimes 1)(1 \otimes d_\otimes)$ (the product rule)

[d.4] $d_\otimes \delta = (1 \otimes \Delta_\otimes) a_\otimes (d_\otimes \otimes \delta) d_\otimes$ (the chain rule)

[d.5] $(1 \otimes d_\otimes) d_\otimes = a_\otimes (c_\otimes \otimes 1) a_\otimes^{-1} (1 \otimes d_\otimes) d_\otimes$ (the interchange rule)

A monoidal*storage* differential category simply means that the storage transformations (see Definition 3.1.1) are isomorphisms. Note also that in [BCS06] a differential category was defined to be one satisfying only the first four of these conditions: the last condition, the interchange rule, was introduced in [BCS09] in order to ensure that the coKleisli category was a Cartesian differential category. Here we shall simply add the interchange law as a condition for being a monoidal differential category, as it will turn out not only to be sufficient but necessary to obtain the characterization of monoidal differential storage categories (with an exact modality) as the linear maps of a Cartesian differential storage category.

The remainder of the section is dedicated to proving Theorem 4.3.1.

Given a Cartesian differential storage category here is the definition of the monoidal deriving transformation:

$$\begin{tikzcd}
A \times S(A) \arrow[r, "\eta\times 1"] \arrow[d, "\varphi_\otimes"'] & S(A) \times S(A) \arrow[r, "m_\times"] & S(A \times A) \arrow[r, "S(D[\varphi_A])"] & S^2(A) \arrow[d, "\epsilon"] \\
A \otimes S(A) \arrow[rrr, dotted, "d_\otimes"'] & & & S(A)
\end{tikzcd}$$

We start by observing:

4.3.2. Lemma.

(i) $d_\otimes$ *is natural for linear maps;*

(ii) The monoidal deriving transformation and the differential are inter-definable with

$$D[f] = (1 \times \varphi)\varphi_\otimes d_\otimes S(f)\epsilon.$$

Proof.

(i) Note that $d_\otimes S(f) = (f \otimes S(f))d_\otimes$ if and only if $\varphi_\otimes d_\otimes S(f) = \varphi_\otimes (f \otimes S(f))d_\otimes$ but $\varphi d_\otimes = (\eta \times 1)m_\times S(D[\varphi])\epsilon$ which is natural for linear f. This provides the result immediately.

(ii) It suffices to prove the equality:

$$\begin{aligned}
&(1 \times \varphi)\varphi_{\otimes} d_{\otimes} S(f)\epsilon \\
&\quad = (1 \times \varphi)(\eta \times 1) m_{\times} S(D[\varphi])\epsilon S(f)\epsilon \\
&\quad = (1 \times \varphi)(\eta \times 1) m_{\times} S(D[\varphi]) S(S(f)\epsilon)\epsilon] \\
&\quad = (\eta \times 1)(1 \times \varphi) m_{\times} S(D[\varphi S(f)\epsilon])\epsilon \\
&\quad = (\eta \times 1)\theta S(D[f\varphi\epsilon])\epsilon \quad \text{(strength)} \\
&\quad = (1 \times \varphi)(1 \times \epsilon) D[f] \quad \text{(linearity of differential in first argument)} \\
&\quad = D[f]
\end{aligned}$$

■

We want to show that $d_{\otimes}$ as defined satisfies **(d.1)**–**(d.5)** above. So we shall simply go through the conditions:

[d.1] Constants ($d_{\otimes} e = 0$)

$$\begin{aligned}
d_{\otimes} e &= d_{\otimes} S(0) s_{\top} \\
&= (0 \otimes S(0)) d_{\otimes} s_{\top} \\
&= 0
\end{aligned}$$

[d.2] Differentials of linear maps ($d_{\otimes}\epsilon = 1 \otimes e$)

$$\begin{aligned}
\varphi_{\otimes} d_{\otimes}\epsilon &= (\eta \times 1) m_{\times} S(D[\varphi])\epsilon\epsilon \\
&= (\eta \times 1) m_{\times} S(D[\varphi]\epsilon)\epsilon \\
&= (\eta \times 1) m_{\times} S(D[\varphi\epsilon])\epsilon \\
&= (\eta \times 1) m_{\times} S(D[1])\epsilon \\
&= (\eta \times 1) m_{\times} S(\pi_0)\epsilon \\
&= (\eta \times 1)\pi_0\epsilon \\
&= \pi_0\eta\epsilon \\
&= \pi_0 = \varphi_{\otimes}(1 \otimes e)
\end{aligned}$$

[d.3] The product rule ($d_{\otimes}\Delta = (1 \otimes \Delta)(d_{\otimes} \otimes 1) + (1 \otimes \Delta)(c_{\otimes} \otimes 1)(1 \otimes d_{\otimes})$)

This is more complicated. Start by noting:

$$\begin{aligned}
d_{\otimes}\Delta &= d_{\otimes} S(\langle 1, 0\rangle + \langle 0, 1\rangle) s_{\otimes} \\
&= ((\langle 1, 0\rangle + \langle 0, 1\rangle) \otimes S(\langle 1, 0\rangle + \langle 0, 1\rangle)) d_{\otimes} s_{\otimes} \\
&= (\langle 1, 0\rangle \otimes S(\langle 1, 0\rangle + \langle 0, 1\rangle)) d_{\otimes} s_{\otimes} \\
&\qquad + (\langle 0, 1\rangle \otimes S(\langle 1, 0\rangle + \langle 0, 1\rangle)) d_{\otimes} s_{\otimes}
\end{aligned}$$

It suffices to prove that

$$(\langle 1,0\rangle \otimes S(\langle 1,0\rangle + \langle 0,1\rangle))d_\otimes s_\otimes = (1\otimes\Delta)(d_\otimes \otimes 1) \qquad \text{and}$$

$$(\langle 0,1\rangle \otimes S(\langle 1,0\rangle + \langle 0,1\rangle))d_\otimes s_\otimes = (1\otimes\Delta)(c_\otimes \otimes 1)(1\otimes d_\otimes)$$

We shall demonstrate the former leaving the latter to the reader. However, we require some observations first:

4.3.3. Lemma.

(i) If $h\colon A\times B \to C$ is bilinear, then $\langle\langle a,b\rangle,\langle c,e\rangle\rangle D[h] = \langle a,e\rangle h + \langle c,b\rangle h$;

(ii) $\langle\langle a,b\rangle,\langle c,e\rangle\rangle D[m_\times] = \langle a,e\rangle m_\times + \langle c,b\rangle m_\times$;

(iii) $\langle\langle a,b\rangle,\langle c,e\rangle\rangle D[\varphi_{A\times B}] = \langle\langle a,c\rangle D[\varphi_A], e\varphi_B\rangle m_\times + \langle c\varphi_A, \langle b,e\rangle D[\varphi_B]\rangle m_\times$;

(iv) $(\langle 1,0\rangle \times 1)D[\varphi_{A\times B}] = a_\times(D[\varphi_A]\times\varphi_B)m_\times$.

Proof.

(i) We have the following calculation:

$$\begin{aligned}
\langle\langle a,b\rangle,\langle c,e\rangle\rangle D[h] &= \langle\langle a,0\rangle,\langle c,e\rangle\rangle D[h] + \langle\langle 0,b\rangle,\langle c,e\rangle\rangle D[h] \\
&= \langle a,\langle c,e\rangle\rangle D_B[h] + \langle b,\langle c,e\rangle\rangle D_A[h] \\
&= \langle a,e\rangle h + \langle c,b\rangle h \qquad (h \text{ is bilinear}).
\end{aligned}$$

(ii) Use the fact that $m_\times$ is bilinear (see Lemma 3.2.4).

(iii) Using the fact that $\varphi_{A\times B} = (\varphi_A\times\varphi_B)m_\times$ (see Proposition 2.3.7) we have:

$$\begin{aligned}
&\langle\langle a,b\rangle,\langle c,e\rangle\rangle D[\varphi_{A\times B}] \\
&\quad= \langle\langle a,b\rangle,\langle c,e\rangle\rangle D[(\varphi_A\times\varphi_B)m_\times] \\
&\quad= \langle\langle a,b\rangle,\langle c,e\rangle\rangle\langle D[\varphi_A\times\varphi_B],\pi_1(\varphi_A\times\varphi_B)\rangle D[m_\times] \\
&\quad= \langle\langle a,b\rangle,\langle c,e\rangle\rangle\langle\langle(\pi_0\times\pi_0)D[\varphi_A],(\pi_1\times\pi_1)D[\varphi_B]\rangle,\langle\pi_1\pi_0\varphi_A,\pi_1\pi_1\varphi_B\rangle\rangle D[m_\times] \\
&\quad= \langle\langle\langle a,c\rangle D[\varphi_A],\langle b,e\rangle D[\varphi_B]\rangle,\langle c\varphi_A,e\varphi_B\rangle\rangle D[m_\times] \\
&\quad= \langle\langle a,c\rangle D[\varphi_A],e\varphi_B\rangle m_\times + \langle c\varphi_A,\langle b,e\rangle D[\varphi_B]\rangle m_\times.
\end{aligned}$$

(iv) We apply (iii):

$$\begin{aligned}
&(\langle 1,0\rangle\times 1)D[\varphi_{A\times B}] \\
&\quad= \langle\langle\pi_0,0\rangle,\langle\pi_1\pi_0,\pi_1\pi_1\rangle\rangle D[\varphi_{A\times B}] \\
&\quad= \langle\langle\pi_0,\pi_1\pi_0\rangle D[\varphi_A],\pi_1\pi_1\varphi_B\rangle m_\times + \langle\pi_1\pi_1\varphi_A,\langle 0,\pi_1\pi_1\rangle D[\varphi_B]\rangle m_\times \\
&\quad= \langle\langle\pi_0,\pi_1\pi_0\rangle D[\varphi_A],\pi_1\pi_1\varphi_B\rangle m_\times + \langle\pi_1\pi_1\varphi_A,0\rangle m_\times \\
&\quad= \langle\langle\pi_0,\pi_1\pi_0\rangle D[\varphi_A],\pi_1\pi_1\varphi_B\rangle m_\times + 0 \\
&\quad= a_\times(D[\varphi_A]\times\varphi_B)m_\times.
\end{aligned}$$

■

We are now ready to calculate. Since both maps are linear we may prefix each side with the universal map which gives monoidal representation; the universal property tells us that the maps are equal if and only if these composites are equal. Here is the calculation:

$$
\begin{aligned}
&\varphi_\otimes(\langle 1,0\rangle \otimes S(\langle 1,0\rangle + \langle 0,1\rangle))d_\otimes s_2 \\
&= (\langle 1,0\rangle \times S(\Delta_\times))\varphi_\otimes d_\otimes s_2 \\
&= (\langle 1,0\rangle \times S(\Delta_\times))(\eta \times 1)m_\times S(D[\varphi_{A\times A}])\epsilon s_2 \\
&= (\eta \times S(\Delta_\times))m_\times S((\langle 1,0\rangle \times 1)D[\varphi_{A\times A}])\epsilon s_2 \\
&= (\eta \times \Delta_\times)(1 \times m_\times)m_\times S(a_\times(D[\varphi_A] \times \varphi_A)m_\times)\epsilon s_2 \quad \text{(Lemma 4.3.3 (iv))} \\
&= (\eta \times \Delta_\times)a_\times(m_\times \times 1)m_\times S(D[\varphi_A] \times \varphi_A)S(m_\times)\epsilon s_2 \\
&= (\eta \times \Delta_\times)a_\times(m_\times \times 1)(S(D[\varphi_A]) \times S(\varphi_A))m_\times S(m_\times)\epsilon s_2 \\
&= (\eta \times \Delta_\times)a_\times(m_\times \times 1)(S(D[\varphi_A]) \times S(\varphi_A))(\epsilon \times \epsilon)m_\times s_2 \quad \text{(Lemma 2.3.6 (ii))} \\
&= (\eta \times \Delta_\times)a_\times(m_\times \times 1)(S(D[\varphi_A]) \times 1)(\epsilon \times 1)\varphi_\otimes \\
&= (1 \times \Delta_\times)a_\times(((\eta \times 1)m_\times S(D[\varphi_A])\epsilon) \times 1)\varphi_\otimes \\
&= (1 \times \Delta_\times)a_\times((\varphi_\otimes d_\otimes) \times 1)\varphi_\otimes \\
&= (1 \times \Delta_\times)a_\times(\varphi_\otimes \times 1)\varphi_\otimes(d_\otimes \otimes 1) \\
&= (1 \times \Delta_\otimes)\varphi_\otimes(d_\otimes \otimes 1) \\
&= \varphi_\otimes(1 \otimes \Delta_\otimes)(d_\otimes \otimes 1)
\end{aligned}
$$

[d.4] To prove the chain rule, $d_\otimes\delta = (1 \otimes \Delta_\otimes)a_\otimes(d_\otimes \otimes \delta)d_\otimes$, we precompose both sides with $(1 \times \varphi)\varphi_\otimes$. As both maps are clearly linear we can then use the universal properties of φ and $\varphi_\otimes$ to conclude the original maps are equal if and only if these composites are:

$$
\begin{aligned}
&(1 \times \varphi)\varphi_\otimes d_\otimes \delta \\
&\quad = \quad (1 \times \varphi)(\eta \times 1)m_\times S(D[\varphi])\epsilon S(\varphi) \\
&\quad = \quad (1 \times \varphi)(\eta \times 1)m_\times S(D[\varphi]S(\varphi))\epsilon \\
&\quad = \quad (\eta \times \varphi)m_\times S(D[\varphi S(\varphi)])\epsilon \\
&\quad = \quad (\eta \times 1)\theta S(D[\varphi\varphi])\epsilon \\
&\quad = \quad (\eta \times 1)(\epsilon \times 1)D[\varphi\varphi] \\
&\quad = \quad D[\varphi\varphi] \\
&(1 \times \varphi)\varphi_\otimes(1 \otimes \Delta_\otimes)a_\otimes(d_\otimes \otimes \delta)d_\otimes \\
&\quad = \quad (1 \times \varphi)(1 \times \Delta_\otimes)\varphi_\otimes a_\otimes(d_\otimes \otimes \delta)d_\otimes \\
&\quad = \quad (1 \times \varphi\Delta_\times)a_\times(\varphi_\otimes \times 1)\varphi_\otimes(d_\otimes \otimes \delta)d_\otimes \\
&\quad = \quad (1 \times \varphi\Delta_\times)a_\times((\varphi_\otimes d_\otimes) \times S(\varphi))\varphi_\otimes d_\otimes \\
&\quad = \quad (1 \times \varphi\Delta_\times)a_\times(((\eta \times 1)m_\times S(D[\varphi])\epsilon) \times S(\varphi))(\eta \times 1)m_\times S(D[\varphi])\epsilon \\
&\quad = \quad (1 \times \Delta_\times(\varphi\times\varphi))a_\times((\eta\times 1)m_\times S(D[\varphi])\epsilon\times S(\varphi))(\eta\times 1)m_\times S(D[\varphi])\epsilon \\
&\quad = \quad (1 \times \Delta_\times)a_\times(((\eta \times \varphi)m_\times S(D[\varphi])\epsilon) \times \varphi S(\varphi))(\eta \times 1)m_\times S(D[\varphi])\epsilon \\
&\quad = \quad (1 \times \Delta_\times)a_\times(((\eta \times 1)\theta S(D[\varphi])\epsilon) \times (\varphi\varphi))(\eta \times 1)m_\times S(D[\varphi])\epsilon \\
&\quad = \quad (1 \times \Delta_\times)a_\times(((\eta \times 1)(\epsilon \times 1)D[\varphi]) \times (\varphi\varphi))(\eta \times 1)m_\times S(D[\varphi])\epsilon \\
&\quad = \quad (1 \times \Delta_\times)a_\times(D[\varphi] \times \varphi)(\eta \times \varphi)m_\times S(D[\varphi])\epsilon \\
&\quad = \quad (1 \times \Delta_\times)a_\times(D[\varphi] \times \varphi)(\eta \times 1)(\epsilon \times 1)D[\varphi] \\
&\quad = \quad (1 \times \Delta_\times)a_\times(D[\varphi] \times \varphi)D[\varphi] \\
&\quad = \quad D[\varphi\varphi]
\end{aligned}
$$

[d.5] Interchange $((1 \otimes d_\otimes)d_\otimes = a_\otimes(c_\otimes \otimes 1)a_\otimes^{-1}(1 \otimes d_\otimes)d_\otimes)$

We may precompose with $(1 \times 1 \times \varphi)(1 \times \varphi_\otimes)\varphi_\otimes$.

$$
\begin{aligned}
&\langle a, b, c\rangle(1 \times 1 \times \varphi)(1 \times \varphi_\otimes)\varphi_\otimes(1 \otimes d_\otimes)d_\otimes \\
&\quad = \quad \langle a, b, c\rangle(1 \times 1 \times \varphi)(1 \times \varphi_\otimes d_\otimes)\varphi_\otimes d_\otimes \\
&\quad = \quad \langle a, b, c\rangle(1 \times 1 \times \varphi)(1 \times (\eta\times 1)m_\times S(D[\varphi])\epsilon)(\eta\times 1)m_\times S(D[\varphi])\epsilon \\
&\quad = \quad \langle a\eta, b\eta, c\rangle 1 \times \theta(S(D[\varphi])\epsilon)m_\times S(D[\varphi])\epsilon \\
&\quad = \quad \langle a\eta, b\eta, c\rangle(1 \times (\epsilon \times 1)D[\varphi])m_\times S(D[\varphi])\epsilon \\
&\quad = \quad \langle a\eta, \langle b, c\rangle D[\varphi]\rangle m_\times S(D[\varphi])\epsilon
\end{aligned}
$$

The identity requires us to show that a and b can be swapped.

Consider the double differential of φ:

$$\begin{aligned}
\langle\langle 0,a\rangle,\langle b,c\rangle\rangle D[D[\varphi]] &= \langle\langle 0,a\rangle,\langle b,c\rangle\rangle D[\varphi S(D[\varphi])\epsilon] \\
&= \langle 0,a\rangle,\langle b,c\rangle\rangle D[\varphi]S(D[\varphi])\epsilon \\
&= \langle 0,a\rangle,\langle b,c\rangle\rangle D[(\varphi\times\varphi)m_\times]S(D[\varphi])\epsilon \\
&= \langle\langle 0,a\rangle,\langle b,c\rangle\rangle D[\varphi\times\varphi],\langle b\varphi,c\varphi\rangle\rangle D[m_\times]S(D[\varphi])\epsilon \\
&= \langle\langle 0,b\rangle D[\varphi],\langle a,c\rangle D[\varphi]\rangle,\langle b\varphi,c\varphi\rangle\rangle D[m_\times]S(D[\varphi])\epsilon \\
&= (\langle\langle 0,b\rangle D[\varphi],c\varphi\rangle m_\times+\langle b\varphi,\langle a,c\rangle D[\varphi]\rangle m_\times)S(D[\varphi])\epsilon \\
&= (0+\langle b\varphi,\langle a,c\rangle D[\varphi]\rangle m_\times)S(D[\varphi])\epsilon \\
&= \langle b\varphi,\langle a,c\rangle D[\varphi]\rangle m_\times S(D[\varphi])\epsilon
\end{aligned}$$

Then we have:

$$\begin{aligned}
&\langle a\varphi,\langle b,c\rangle D[\varphi]\rangle m_\times S(D[\varphi])\epsilon \\
&= \langle 0,a\rangle,\langle b,c\rangle\rangle D[D[\varphi]] \\
&= \langle 0,b\rangle,\langle a,c\rangle\rangle D[D[\varphi]] \\
&= \langle b\varphi,\langle a,c\rangle D[\varphi]\rangle m_\times S(D[\varphi])\epsilon
\end{aligned}$$

We are nearly there except we would like to change φ into η, which is allowed by another lemma.

4.3.4. Lemma.

$$\langle b\varphi,\langle a,c\rangle D[\varphi]\rangle m_\times S(D[\varphi])\epsilon=\langle a\eta,\langle b,c\rangle D[\varphi]\rangle m_\times S(D[\varphi])\epsilon$$

To prove this lemma we partially differentiate both sides of the equation derived above in the form:

$$\langle\pi_0\pi_0\varphi,\langle\pi_0\pi_1,\pi_1\rangle D[\varphi]\rangle m_\times S(D[\varphi])\epsilon=\langle\pi_0\pi_1\varphi,\langle\pi_0\pi_0,\pi_1\rangle D[\varphi]\rangle m_\times S(D[\varphi])\epsilon$$

with respect to the first coordinate at position 0. This is best done using the term logic. In the derivation below we flip between the term logic and the categorical term. The derivation may best be read from the middle outward:

$$
\begin{aligned}
&\langle \pi_0\pi_0\eta, \langle \pi_0\pi_1, \pi_1\rangle D[\varphi]\rangle m_\times S(D[\varphi])\epsilon \\
&= \quad [\![((a,b),c) \mapsto \epsilon(S(D[\varphi])(m_\times(\eta(a), D[\varphi](c)\cdot b)))]\!] \\
&= \quad \left[\!\!\left[((a,b),c) \mapsto \epsilon(S(D[\varphi])(m_\times(\frac{\partial\varphi(x)}{\partial x}(0)\cdot a, D[\varphi](c)\cdot b)))\right]\!\!\right] \\
&= \quad \left[\!\!\left[((a,b),c) \mapsto \frac{\partial\epsilon(S(D[\varphi])(m_\times(\varphi(x), D[\varphi](c).b)))}{\partial x}(0)\cdot a\right]\!\!\right] \\
&= \quad \langle\langle \pi_0\pi_0, 0, \rangle, 0\rangle, \langle\langle 0, \pi_0\pi_1\rangle, \pi_1\rangle\rangle D[\langle \pi_0\pi_0\varphi, \langle \pi_0\pi_1, \pi_1\rangle D[\varphi]\rangle m_\times S(D[\varphi])\epsilon \\
&= \quad \langle\langle \pi_0\pi_0, 0, \rangle, 0\rangle, \langle\langle 0, \pi_0\pi_1\rangle, \pi_1\rangle\rangle D[\langle \pi_0\pi_1\varphi, \langle \pi_0\pi_0, \pi_1\rangle D[\varphi]\rangle m_\times S(D[\varphi])\epsilon \\
&= \quad \left[\!\!\left[((a,b),c) \mapsto \frac{\partial\epsilon(S(D[\varphi])(m_\times(\varphi(b), D[\varphi](c).x)))}{\partial x}(0)\cdot a\right]\!\!\right] \\
&= \quad \left[\!\!\left[((a,b),c) \mapsto \epsilon(S(D[\varphi])(m_\times(\varphi b, \frac{\partial D[\varphi](c)\cdot x}{\partial x}(0)\cdot a)))\right]\!\!\right] \\
&= \quad [\![((a,b),c) \mapsto \epsilon(S(D[\varphi])(m_\times(\varphi b, D[\varphi](c)\cdot a)))]\!] \\
&= \quad \langle \pi_0\pi_1\varphi, \langle \pi_0\pi_0, \pi_1\rangle D[\varphi]\rangle m_\times S(D[\varphi])\epsilon
\end{aligned}
$$

The key step uses the fact that $\eta(a)$ is the partial derivative of $\varphi(x)$ at 0 with linear argument a, that is $\eta(a) = \frac{\partial\varphi(x)}{\partial x}(0)\cdot a$. Otherwise we are using the fact that we are differentiating in linear positions so the term does not change.

4.4. Differential λ-calculus. Cartesian closed differential categories, as defined in [BEM10, Man12], provide a semantics for the differential and the resource λ-calculus. Cartesian closed differential storage categories, by the results of Section 3.5, are precisely those semantic settings which arise as the coKleisli categories of monoidal closed differential storage categories. However, our definition of a Cartesian closed differential storage category is not transparently a Cartesian closed differential category, in the sense of [BEM10, Man12], as in those references it is required that differentiation can be moved inside the λ-abstraction. Recall that here we have only demanded the linearity of evaluation in higher-order arguments. In fact, we shall show these conditions are equivalent and hence the following proposition.

4.4.1. Proposition. *To have a closed differential storage category is precisely to have a Cartesian closed differential category whose linear maps are (strongly and persistently) classified and which has linear hom-objects.*

Proof. To establish this it suffices to observe that linearity of evaluation in its higher-order argument and being able to move differentials inside λ-abstractions are equivalent requirements. We start be assuming the latter:

Using the term logic, the ability to move differential under the λ-abstraction may be expressed as

$$\frac{\partial\lambda x.t}{\partial y}(p)\cdot v = \lambda x.\left(\frac{\partial t}{\partial y}(p)\cdot v\right)$$

A consequence of this requirement, as proven for example in [Man12, Lemma 3.10], (here translated into the term logic), is that

$$\frac{\partial t\, s}{\partial x}(p)\cdot v = \frac{\partial t[p/x]\, y}{\partial y}(s[p/x])\cdot\left(\frac{\partial s}{\partial x}(p)\cdot v\right) + \left(\frac{\partial t}{\partial x}(p)\cdot v\right) s[p/x]$$

and so, if s does not contain x (so that $\frac{\partial s}{\partial x}(p)\cdot v = 0$ and $s[p/x] = s$)

$$\frac{\partial t\, s}{\partial x}(p)\cdot v = \left(\frac{\partial t}{\partial x}(p)\cdot v\right) s$$

which amounts to the observation that evaluation is linear in its higher-order argument.

Conversely, recall (Definition 2.2.2) that a basic assumption on a closed system of maps is that evaluation should be linear in its higher-order argument. So, by the following argument (using η expansion) we see that one can move differentiation inside a λ-abstraction:

$$\frac{\partial \lambda x.t}{\partial y}(p)\cdot v = \left[\lambda x.\left(\frac{\partial \lambda x.t}{\partial y}(p)\cdot v\right)\right] x = \lambda x.\left(\frac{\partial(\lambda x.t)\, x}{\partial y}(p)\cdot v\right) = \lambda x.\left(\frac{\partial t}{\partial y}(p)\cdot v\right)$$

Thus, the axiom that states that one can move differentiation inside λ-abstractions is actually equivalent (under the adjunction for closure) to the requirement that evaluation be linear in its higher-order argument. ∎

This allows us to establish:

4.4.2. Theorem. *The coKleisli category of a monoidal differential storage category (satisfying the interchange rule) is a Cartesian closed differential storage category. Furthermore, the linear maps of a Cartesian closed differential storage category, in which linear idempotents split, form a monoidal closed differential storage category.*

Proof. For the first statement: Proposition 3.5.4 tells us that the coKleisli category of a monoidal closed storage category is a Cartesian closed storage category. Applying this to the case of a monoidal closed *differential* storage category produces a Cartesian closed differential storage category, since the linear maps form a closed system and evaluation is linear in its higher-order argument.

For the second statement we note that by Definition 3.5.1, being a Cartesian closed differential storage category not only requires that the setting is Cartesian closed and that the linear maps form a closed system, but also that the equalizer for forming the object of linear maps is present. This then means the linear maps are a monoidal closed differential category by Corollary 3.5.3 and Theorem 4.3.1. ∎

References

[B95] Gavin Bierman *What is a categorical model of intutionistic linear logic.* LNCS **92** (1995) 73–93.

[BET12] R. Blute, T. Ehrhard and C. Tasson *A Convenient Differential Category.* Cahiers de Topologie et Géométrie Différentielle Catégorique, **53**. (2012) 211-233.

[BCS06] R. Blute, J.R.B. Cockett, R.A.G. Seely *Differential categories.* Mathematical Structures in Computer Science **16** (2006) 1049–1083.

[BCS09] R. Blute, J.R.B. Cockett, R.A.G. Seely *Cartesian differential categories.* Theory and Applications of Categories **22** (2009) 622–672.

[BEM10] A. Bucciarelli, T. Ehrhard, and G. Manzonetto *Categorical Models for Simply Typed Resource Calculi.* Electronic Notes in Theoretical Computer Science, vol. 265 (2010) 213-230.

[CS11] J.R.B. Cockett, R.A.G. Seely *The Faà di Bruno Construction.* Theory and Applications of Categories **25** (2011) 392-425.

[E04] T. Ehrhard *Finiteness spaces.* Mathematical Structures in Computer Science **15** (2005) 615–646.

[ER04] T. Ehrhard, L. Regnier *The differential λ-calculus.* Theoretical Computer Science, 309(1-3) (2003) 1–41.

[ER05] T. Ehrhard, L. Regnier *Differential interaction nets.* Workshop on Logic, Language, Information and Computation (WoLLIC), invited paper. Electronic Notes in Theoretical Computer Science, vol. 123, March 2005, Elsevier.

[F07] M.P. Fiore *Differential structure in models of multiplicative biadditive intuitionistic linear logic.* In Typed Lambda Calculi and Applications, Lecture Notes In Computer Science 4583. Springer-Verlag (2007) 163–177.

[F99] C. Fuhrmann *Directe models of the computational lambda calculus.* ENTCS **20** (1999) 147–172.

[J02] P.T. Johnstone, *Sketches of an Elephant: a topos theory compendium* **1**, Oxford University Press, 2002.

[KE65] G.M. Kelly, S. Eilenberg *Closed categories.* In Proceedings of the Conference on Categorical Algebra, La Jolla, 1965, Berlin, Springer, 1966.

[Kock12] A. Kock *Commutative monoids as a theory of distributions.* Theory and Applications of Categories **26** (2012) 97–131.

[G87] J.-Y. Girard *Linear logic.* Theoretical Computer Science **50** (1987) 1–102.

[Man12] G. Manzonetto *What is a categorical model of the differential and the resource λ-calculi?* Mathematical Structures in Computer Science 22(3) (2012) 451-520.

[M02] Paul-André Melliés *Categorical models of linear logic revisited.* Prépublication de l'équipe PPS, September (2003) number 22.

[S04] Andrea Schalk *What is a categorical model for linear logic?* (2004).

[S89] R.A.G. Seely *Linear logic, $*$-autonomous categories and cofree coalgebras.* AMS Contemporary Mathematics **92** (1989) 371–382.

[Street] R. Street *Skew-closed categories*, Journal of Pure and Applied Algebra 217(6) (June 2013) 973-988.

Department of Mathematics, University of Ottawa,
585 King Edward St., Ottawa, ON, K1N 6N5, Canada

Department of Computer Science, University of Calgary,
2500 University Drive, Calgary, AL, T2N 1N4, Canada

Department of Mathematics and Statistics, McGill University,
805 Sherbrooke St., Montréal, QC, H3A 2K6, Canada

Email: `rblute@uottawa.ca`
`robin@ucalgary.ca`
`rags@math.mcgill.ca`

This article may be accessed at `http://www.tac.mta.ca/tac/`

THEORY AND APPLICATIONS OF CATEGORIES (ISSN 1201-561X) will disseminate articles that significantly advance the study of categorical algebra or methods, or that make significant new contributions to mathematical science using categorical methods. The scope of the journal includes: all areas of pure category theory, including higher dimensional categories; applications of category theory to algebra, geometry and topology and other areas of mathematics; applications of category theory to computer science, physics and other mathematical sciences; contributions to scientific knowledge that make use of categorical methods.

Articles appearing in the journal have been carefully and critically refereed under the responsibility of members of the Editorial Board. Only papers judged to be both significant and excellent are accepted for publication.

Full text of the journal is freely available in .dvi, Postscript and PDF from the journal's server at `http://www.tac.mta.ca/tac/` and by ftp. It is archived electronically and in printed paper format.

SUBSCRIPTION INFORMATION Individual subscribers receive abstracts of articles by e-mail as they are published. To subscribe, send e-mail to `tac@mta.ca` including a full name and postal address. For institutional subscription, send enquiries to the Managing Editor, Robert Rosebrugh, `rrosebrugh@mta.ca`.

INFORMATION FOR AUTHORS The typesetting language of the journal is TEX, and LATEX2e strongly encouraged. Articles should be submitted by e-mail directly to a Transmitting Editor. Please obtain detailed information on submission format and style files at `http://www.tac.mta.ca/tac/`.

MANAGING EDITOR. Robert Rosebrugh, Mount Allison University: `rrosebrugh@mta.ca`

TEXNICAL EDITOR. Michael Barr, McGill University: `barr@math.mcgill.ca`

ASSISTANT TEX EDITOR. Gavin Seal, Ecole Polytechnique Fédérale de Lausanne: `gavin_seal@fastmail.fm`

TRANSMITTING EDITORS.

Clemens Berger, Université de Nice-Sophia Antipolis: `cberger@math.unice.fr`
Richard Blute, Université d' Ottawa: `rblute@uottawa.ca`
Lawrence Breen, Université de Paris 13: `breen@math.univ-paris13.fr`
Ronald Brown, University of North Wales: `ronnie.profbrown(at)btinternet.com`
Valeria de Paiva: `valeria.depaiva@gmail.com`
Ezra Getzler, Northwestern University: `getzler(at)northwestern(dot)edu`
Kathryn Hess, Ecole Polytechnique Fédérale de Lausanne: `kathryn.hess@epfl.ch`
Martin Hyland, University of Cambridge: `M.Hyland@dpmms.cam.ac.uk`
Anders Kock, University of Aarhus: `kock@imf.au.dk`
Stephen Lack, Macquarie University: `steve.lack@mq.edu.au`
F. William Lawvere, State University of New York at Buffalo: `wlawvere@buffalo.edu`
Tom Leinster, University of Edinburgh: `Tom.Leinster@ed.ac.uk`
Ieke Moerdijk, Radboud University Nijmegen: `i.moerdijk@math.ru.nl`
Susan Niefield, Union College: `niefiels@union.edu`
Robert Paré, Dalhousie University: `pare@mathstat.dal.ca`
Jiri Rosicky, Masaryk University: `rosicky@math.muni.cz`
Giuseppe Rosolini, Università di Genova: `rosolini@disi.unige.it`
Alex Simpson, University of Edinburgh: `Alex.Simpson@ed.ac.uk`
James Stasheff, University of North Carolina: `jds@math.upenn.edu`
Ross Street, Macquarie University: `street@math.mq.edu.au`
Walter Tholen, York University: `tholen@mathstat.yorku.ca`
Myles Tierney, Rutgers University: `tierney@math.rutgers.edu`
R. J. Wood, Dalhousie University: `rjwood@mathstat.dal.ca`