Dialectica Comonads

Valeria de Paiva
Topos Institute, Berkeley, CA, USA

Abstract

Dialectica categories are interesting categorical models of Linear Logic which preserve the differences between linear connectives that the logic is supposed to make, unlike some of the most traditional models like coherence spaces. They arise from the categorical modelling of Gödel's Dialectica interpretation and seem to be having a revival: connections between Dialectica constructions and containers, lenses and polynomials have been described recently in the literature. In this note I will recap the basic Dialectica constructions and then go on to describe the less well-known interplay of comonads, coalgebras and comonoids that characterizes the composite functor standing for the "of course!" operator in dialectica categories. This composition of comonads evokes some work on stateful games by Laird and others, also discussed in the setting of Reddy's system LLMS (Linear Logic Model of State).

2012 ACM Subject Classification Theory of computation → Logic

Keywords and phrases Dialectica categories, Linear logic, Comonads

Digital Object Identifier 10.4230/LIPIcs.CALCO.2021.3

Category Invited Talk

Funding *Valeria de Paiva*: Research supported by AFOSR grant FA9550-20-10348.

Acknowledgements I would like to thank Fabio Gadducci and Alexandra Silva for the invitation to speak at CALCO2021. Thinking about the CALCO community and how it influences my work was much more fun than I had expected.

1 Introduction

Monads and comonads, T-algebras and F-coalgebras have been the subject of much investigation within the CALCO (Conference on Algebra and Coalgebra in Computer Science) community, since at least 2005, when the conferences CMCS (Workshop on Coalgebraic Methods in Computer Science) and WADT (Workshop on Algebraic Development Techniques) joined forces to form CALCO. Even earlier on, in 1998, Jacobs, Moss, Reichel and Rutten already wrote in the preface to the first CMCS proceedings that "Coalgebra is beginning to develop into a field of its own, with its own model theory and proof methods (involving bisimulations and invariants)."

The equiparation between algebraic methods and coalgebraic ones is a relatively recent phenomenon, as algebraic methods were much more prevalent in mathematics to begin with. Also in computing the notions of monads and algebras had a head start both with algebraic data types in the eighties and with notions of "computation as monads" in Moggi's and others work in the nineties. Part of the historic phenomenon was indeed that, because categorical dualities are so ubiquitous, many researchers looked at their own tools and results and considered "reversed arrows". This means that the work in coalgebraic methods comes from very different areas and motivations.

Our motivation here is clearly logical, coming from the categorical modelling of (especially intuitionistic) linear and modal logics. We describe quickly Linear Logic, then its models based on Gödel's Dialectica interpretation, then the difficulties of modelling what I take to be

the main innovation of linear logic (the shriek or bang modality $!A$). Finally, we discuss how the models devised (the dialectica spaces) turned out to model other interactions between operators that seem relevant in computing.

2 Linear Logic

Linear logic is a logic proposed by Jean-Yves Girard as a refinement of both classical and intuitionistic logic more than 30 years ago. Linear logic possesses the dualities of classical logic, but also many of the constructive properties of intuitionistic logic. Linear logic lends itself to many different presentations, explanations, and intuitions. Proof-theoretically, it derives from an analysis of classical sequent calculus in which uses of (the structural rules) contraction and weakening are carefully controlled. Operationally, this means that logical deduction is no longer merely about a collection of persistent "truths", but also a way of manipulating resources that cannot always be duplicated or thrown away at will.

In terms of denotational models, linear logic may be seen as refining the interpretation of intuitionistic logic by replacing cartesian (closed) categories by symmetric monoidal (closed) categories. But this is partial, and somewhat misleading because, while models of intuitionistic propositional logic are specific kinds of cartesian closed categories, models of linear logic need to take into account how linear logic relates to intuitionistic logic, so they are better conceived (as has been the case recently) as "symmetric monoidal adjunctions" between a cartesian closed category (modelling the intuitionistic world) and a symmetric monoidal closed category, modelling the linear world, following the work of Benton [5] and others.

This adjunction can be seen as a monoidal comonad (the operator !) in the linear category $\mathcal{L}$ (satisfying some extra properties), or as a monad in the intuitionistic category $\mathcal{C}$, again with extra properties. This monoidal comonad ! behaves very much like a constructive necessity operator $\Box$ in (S4) modal logic, so the whole area of type theory and categorical semantics for Modal Logics can be given another (co)algebraic treatment, different from the usual one, which is based on generalizing its Kripke-like semantics. This kind of type theory and categorical semantics for modal logics can then be used as a blueprint to investigate other modal-like languages, such as hybrid logics, description logics, temporal logics, etc.

A warning is that when we talk about logics, we are considering preferentially constructive or intuitionistic logics. While we would like to sidestep as much as possible philosophical questions about the nature of mathematics, of logic and computation, etc., it seems clear that constructive reasoning principles are safer. If one asks whether "is there an x such that $P(x)$ holds?" most of us will be happier with the answer "yes, x_0", than with an answer of the form "yes, for all x it is not the case that not $P(x)$". We want reasoning to be as precise and safe as possible. Thus, we would like to use constructive reasoning as much as possible. If one needs a classical principle, we hope to be able to flag it, and to know where and why it is needed. Hence, we consider in this note (co)monads and (co)algebras as **constructive modalities** in an extended view of the categorical Curry-Howard correspondence.

2.1 Modalities

Modalities are unary operators over some logic basis. The logical basis can be a classical, an intuitionistic or a linear basis. A logician can think of themselves as exploring the landscape of possible logical systems, trying to answer questions such as: Which modalities? Which basis? Why do I need a modality? How do I choose between modalities?

Axioms:

$$A \to (B \to A)$$
$$(A \to (B \to C)) \to ((A \to B) \to (A \to C))$$
$$A \to (B \to A \wedge B)$$
$$A \wedge B \to A$$
$$A \wedge B \to B$$
$$(A \to C) \to ((B \to C) \to (A \vee B \to C))$$
$$A \to A \vee B$$
$$B \to A \vee B$$
$$\bot \to A$$

Figure 1 Axioms for propositional intuitionistic logic.

To start to answer these questions, we remind the reader that logics can be presented in several different styles. The most traditional formal system to describe a logic is a collection of axioms, in a Hilbert style calculus. For example Figure 1 describes an axiomatization of the intuitionistic propositional calculus. This axiomatization relies on a single rule, Modus Ponens:

$$\frac{\Rightarrow A \to B \qquad \Rightarrow A}{\Rightarrow B}\ \textit{Modus Ponens}$$

(plus substitutions or axiom-schemes)

The main advantage of axiomatic systems are that it is easier to prove theorems *about* a logic described using them. The main problem with axiomatic systems is that it is difficult to prove theorems *within* the logic, but also that there is no canonical system of axioms. The axioms can be true for very obscure reasons. There are several choices of axioms and no obvious criterion to choose between the possibilities.

Gentzen systems of sequent calculus and natural deduction are much more reliable. Sequent calculus systems consist of structural and logical rules. For each logical connective we have left and right rules. The main advantage of sequent calculi is that it is easier to find proofs in the system, which facilitates automated deduction. The system is very symmetrical, and the geometry of derivations helps to get theorems about the working of the system itself.

Natural Deduction is the other Gentzen system for proof construction and natural deduction formulations of logic are usually preferred by logicians interested in proofs. These model better how humans reason, hence the adjective "natural". Many proofs assistants use Natural Deduction, either in Prawitz or in Martin-Löf style, as their formalism to describe logics.

Looking at presentations of the sequent calculus we can see the cleverness behind Girard's Linear Logic. Girard removed the structural rules of contraction and weakening from usual rules of logic, so the logic becomes resource aware. One cannot discard premises not used in a proof (weakening), one cannot duplicate a premise A that we used once (contraction).

$$\frac{\Gamma \Rightarrow C}{\Gamma,\, A \Rightarrow C}\ \textit{Weakening}$$

$$\frac{\Gamma,\, A,\, A \Rightarrow C}{\Gamma,\, A \Rightarrow C}\ \textit{Contraction}$$

However, if you simply remove these two structural rules, the logic is too weak. To get back the expressive power of usual logic, Girard uses a modality, written as ! and called "of course!". This works mostly via the translation

$$A \to B = (!A) \multimap B.$$

Contraction and weakening are available only for $!A$ formulas/objects. The operator ! (shriek or bang) is a unary operator over a linear basis whose rules correspond to the $\Box$ (or necessity) S4 modality. It was realized very early on that this modality should be modelled categorically by a comonad.

Why? Which kind of comonad? Before facing these questions, we expand our description of a propositional linear basis and its categorical modelling.

2.2 The challenges of modeling Linear Logic

When modelling of Linear Logic is discussed the first task is to explain the difference between a resource-aware implication and an usual one. A traditional implication satisfies:

$A, A \to B \ \vdash\ B$
$A, A \to B \ \vdash\ A \wedge B$

Thus one can use Modus Ponens (MP) to obtain B, from A and $A \to B$, but we still have A after completing the inference. Meanwhile a linear implication satisfies instead:

$A, A \multimap B \ \vdash B$
$A, A \multimap B \ \nvdash A \otimes B$

As before we can deduce B from A and $A \multimap B$, but A is then gone, we cannot produce it again, from nowhere. Similarly, the traditional conjunction $A \wedge B$ satisfies $A \wedge B \vdash B$ (or A) while a linear conjunction does not entail its conjuncts $A \otimes B \ \nvdash A$. We cannot project A (or B) from a linear conjunction.

When we add the modality ! we can of course re-use formulae and discard them, if they have the right type.

Of course:	$!A \ \vdash\ !A \otimes !A$	Re-use
	$!A \otimes B \ \vdash\ I \otimes B \cong B$	Discard

The traditional categorical modeling of intuitionistic logic, associates a formula A to an object A of an appropriate (i.e. a cartesian closed) category $\mathbb{C}$. The intuitionistic conjunction $A \wedge B$ is associated to a cartesian product $A \times B$ and the intuitionistic implication $A \to B$ is associated to a function space B^A (the homset in $\mathbb{C}$ of functions from A to B).

The usual conjunction are modelled by real cartesian products, so we have projections $(A \times B \to A, B)$, and diagonals $(A \to A \times A)$, which correspond to deletion and duplication of resources. This is not a linear setting. To cope with the removal of the structural rules of weakening and contraction, the structure needs to be relaxed from a cartesian closed category to a monoidal closed category, with tensors and internal-homs. This move is not problematic, as category theorists had the mathematics done several decades before they were needed. We may also need to have coproducts and products to model the linear logic additives (written as $\oplus$ and &), if these are desired. But these structures were also part of the category theorists vocabulary and their adaptation to the then new setting was not problematic.

However, the main challenge of modelling linear logic was really the operator $!A$. Let us recap its rules:

$$\frac{!\Gamma \vdash B}{!\Gamma \vdash !B} \qquad \frac{\Gamma \vdash B}{\Gamma, !A \vdash B} \qquad \frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B} \qquad \frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}$$

If in the first rule above we substitute $!\Gamma$ for $!A$ and B for $!A$, we end up with

$$\frac{!A \vdash !A}{!A \vdash !!A}$$

which shows that we need to have a morphism in the category with shape $\delta\colon !A \to !!A$, as any category has the identity $id_{!A}\colon !A \to !A$. Similarly, for the other rules below:

$$\frac{\Gamma = !A, B = !A}{\delta\colon !A \to !!A} \qquad \frac{\Gamma = \emptyset, B = I}{\mathsf{er}\colon !\mathsf{A} \to \mathsf{I}} \qquad \frac{\Gamma = \emptyset, B = A}{\mathsf{der}\colon !\mathsf{A} \to \mathsf{A}} \qquad \frac{\Gamma = \emptyset, B = !A \otimes !A}{\mathsf{copy}\colon !\mathsf{A} \to !\mathsf{A} \otimes !\mathsf{A}}$$

Thus we see that, for each object of the category $\mathbb{C}$ of the form $!A$ we must have a collection of natural morphisms:

$$!A \xrightarrow{\mathsf{prom}} !!A$$

$$!A \xrightarrow{\mathsf{counit}} I$$

$$!A \xrightarrow{\mathsf{der}} A$$

$$!A \xrightarrow{\mathsf{copy}} !A \otimes !A$$

Then we infer that $!A$ needs to be a commutative comonoid with respect to the tensor product. But also that it should be a coalgebra for the monoidal comonad "!". The comonoid and the coalgebra structures need to interact in a consistent and coherent way, which makes many equations required. Which is the best way to describe these interactions?

3 Dialectica spaces

Historically Seely [28], Lafont [17] and de Paiva [9] presented their categorical models more or less at the same time, around 1987. But de Paiva investigated a particular, concrete case of Seely's generic idea. Her original model also happens to be an instance of Lafont's cofree model.

Seely isomorphisms, already present in Girard's original paper, mix multiplicative and additive categorical structures.

$$!(A \& B) \cong !A \otimes !B \text{ and } !1 \cong I$$

Work by Benton et al [4] and Bierman [6] came with a solution of model (for categories without products), but the solution involved too many "commuting conversions". Benton's LNL system [5] and Barber's DILL "Dual Intuitionistic and Linear Logic"[2], as well as Maietti et al [23] "Relating Models of Intuitionistic Linear Logic" and Hyland and Schalk's [20] glueing models have a working theory of what models should look like. A more complete discussion, with full proofs, but from a very different perspective, can be found in Melliès [1].

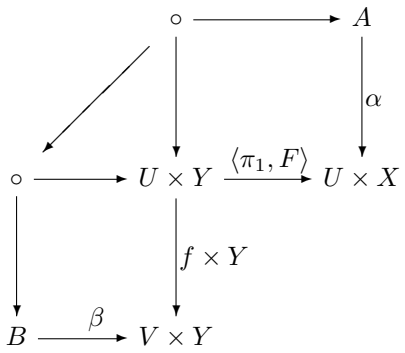
3.1 Dialectica Spaces

The Dialectica constructions give us a precision-based model of Linear Logic. All connectives correspond to distinct structures in the categories. We describe two families of models: the original Dialectica categories $Dial_2(\mathbb{C})$ model intuitionistic linear logic, while the Dialectica-like construction, suggested by Girard, the categories $DDial_2(\mathbb{C})$, model classical or full intuitionistic linear logic. We use the expression *dialectica spaces* to refer to both constructions.

The difference between these constructions is mostly on morphisms, objects are the same, i.e. triples $(U, X, \alpha : A \hookrightarrow U \times X)$ where U, X are objects of $\mathbb{C}$, a cartesian closed category and α a relation between the objects U, X . However, different notions of morphism will produce very different structures in the resulting categories. However, both constructions produce dialectica categories $Dial_2(\mathbb{C})$ and $DDial_2(\mathbb{C})$ which are symmetric monoidal closed categories with products, as we recall below.

The first construction, $Dial_2(\mathbb{C})$, comes from the internalization of the functionals in Gödel's Dialectica interpretation. Given a cartesian closed category $\mathbb{C}$, we construct the category $Dial_2(\mathbb{C})$ as follows. The objects of $Dial_2(\mathbb{C})$ are relations between U and X, so an object in $Dial_2(\mathbb{C})$ is a monic $A \overset{\alpha}{\hookrightarrow} U \times X$ written as $(U \overset{\alpha}{\leftarrow\!\!+} X)$.

A map in $\mathsf{Dial}_2(C)$ between objects $A = (U, X, \alpha)$ and $B = (V, Y, \beta)$ is a pair of maps in C, $(f\colon U \to V, F\colon U \times Y \to X)$ satisfying the pullback condition below. We can think of this condition in **Set** as saying if $u\alpha F(u, y)$ then $f(u)\beta y$. Thus



This is where the mathematics gets somewhat involved, as calculating pullbacks can be tiresome. One has to prove that the definitions make sense and moreover that they work. That is, the category $Dial_2(\mathbb{C})$ with the structure described below is a symmetric monoidal closed category with products. The tensor product, the internal-hom and the categorical product in $Dial_2(\mathbb{C})$ are given as follows:

$$A \oslash B = (U \times V \overset{\alpha \oslash \beta}{\leftarrow\!\!+} X \times Y)$$

$$A \multimap B = (V^U \times X^{U \times Y} \overset{\alpha \multimap \beta}{\leftarrow\!\!+} U \times Y)$$

$$A \& B = (U \times V \overset{\alpha\beta}{\leftarrow\!\!+} X + Y)$$

The category has only *weak coproducts* and a weak initial object. The full calculations and properties needed to show that this structure works as required can be found in the technical report [14].

▶ **Theorem 1** (de Paiva 1987). *The category $Dial_2(\mathbb{C})$ has a symmetric monoidal closed structure (and products and weak coproducts), that makes it a model of (exponential-free)* **intuitionistic** *linear logic.*

The category $Dial_2(\mathbb{C})$ is a good model for Linear Logic, since it has the structure needed to model intuitionistic linear logic, including a highly non-trivial and very well-behaved modality !. The model was an independent confirmation of the worth of Linear Logic, since it is based on Gödel's Dialectica interpretation, an interesting, useful and still a bit mysterious tool in proof theory [21]. However, it was not clear if there was an obvious way of transforming this model into a model of Classical Linear Logic.

We can reformulate $Dial_2(\mathbb{C})$ a little: Say we consider the category $\mathbb{C}$ as **Set**, objects are maps $U \times X \xrightarrow{\alpha} 2$, morphisms are pairs of maps $f\colon U \to V$, $F\colon U \times Y \to X$ such that in the diagram

$$\begin{array}{ccc} U \times Y & \xrightarrow{\langle \pi_1,\, F\rangle} & U \times X \\ {\scriptstyle f\times id_Y}\downarrow & & \downarrow{\scriptstyle \alpha} \\ V \times Y & \xrightarrow[\beta]{} & 2 \end{array}$$

we have $\alpha(u,\, F(u,y)) \leq \beta(fu, y)$. The point of this formulation is to compare the categories with other proposed models of linear logic like *Chu spaces* [24].

Following a suggestion of Girard a substantial modification can be made to the category $Dial_2(\mathbb{C})$ to obtain a model of Classical Linear Logic. Actually, in the general case, the category we obtain is a model of Full Intuitionistic Linear Logic [19], but if desired we can restrict ourselves to a model of classical linear logic.

Again we assume that the category $\mathbb{C}$ is a cartesian closed category. The objects of the category $DDial_2(\mathbb{C})$ are relations between U and X in $\mathbb{C}$, as before. However, the maps of $DDial_2(\mathbb{C})$ are now simply inverse pairs of maps of $\mathbb{C}$, $f\colon U \to V$, $F\colon Y \to X$ such that $u\alpha F(y) \Rightarrow f(u)\beta y$. Thus the only difference is that in $Dial_2(\mathbb{C})$, the "counterexample" map F has domain $U \times Y$, while in $DDial_2(\mathbb{C})$, F has domain simply Y.

This simplification of the notion of morphism gives us more structure still to describe. But the work with the categories $Dial_2(\mathbb{C})$ helps to make intuitions clearer. Summarizing the structure in $DDial_2(\mathbb{C})$ is given by:

$$\begin{aligned} A \multimap B &= (V^U \times X^Y \xleftarrow[+]{\alpha \multimap \beta} U \times Y) \\ A \otimes B &= (U \times V \xleftarrow[+]{\alpha \otimes \beta} X^V \times Y^U) \\ A \parr B &= (V^X \times U^Y \xleftarrow[+]{\alpha \parr \beta} X \times Y) \\ A \,\&\, B &= (U \times V \xleftarrow[+]{\alpha \cdot \beta} X + Y) \\ A \oplus B &= (U + V \xleftarrow[+]{\alpha \cdot \beta} X \times Y) \end{aligned}$$

For $Dial_2(\mathbb{C})$ the tensor product was easy to obtain, and we worked what the internal-hom should be, to obtain the adjunction we wanted. For $DDial_2(\mathbb{C})$ the opposite process occurs: we now know the shape of the internal-hom and have to guess an appropriate tensor-product to obtain the monoidal closure. The guessing is shown to work, by proving the adjunction that says that the category is symmetric monoidal closed.

▶ **Theorem 2** (de Paiva 1988). *The category $DDial_2(\mathbb{C})$ has a symmetric monoidal closed structure (and products, coproducts and units), that makes it a model of (exponential-free)* **classical** *linear logic.*

Products and coproducts work as expected. Also the units now work as expected, we have four distinct units and calculations can be found, as before, in the technical report:

$$\begin{aligned} I &= (1 \xleftarrow{+}{}^{i} 1) \\ \bot &= (1 \xleftarrow{+}{}^{e} 1) \\ 1 &= (1 \xleftarrow{+}{}^{\cdot} 0) \\ 0 &= (0 \xleftarrow{+}{}^{\cdot} 1) \end{aligned}$$

Having prepared the linear logic basis (twice!), we now go back to the more interesting issue of the modalities, that is our comonads and their coalgebras and comonoids.

4 Dialectica Comonads

We want *all* the expressive power of traditional logic. For that Linear Logic introduces the modalities ! and ?, which are dual to each other. The intuition for these modalities is well-known: $!A$ means we can use A as many times as we want and, dually, the operator $?A$ means we can create as many A's as desired.

In this section, we first define a modality ! in $Dial_2(\mathbb{C})$ and then, to define the operator "!" for $DDial_2(\mathbb{C})$, we have to compose comonads. We abstract a few lessons from this process.

4.1 A cofree !-comonad

Surprisingly enough the ! operator in $Dial_2(\mathbb{C})$ is a *cofree* comonad. There are many useful and well-known monads in the literature like exceptions, side-effects, powerset, continuations, etc... but there are fewer well-known and loved comonads. Since we want a comonad that creates comonoids with respect to the tensor product in the category $Dial_2(\mathbb{C})$, we want an object $!A$ such that it has a diagonal like function to $!A \otimes !A$. Now whatever the ! operator is, the tensor product $!A \otimes !A$ needs to have something like $U \times U$ as its left-handed side. This is easy, as $\mathbb{C}$ is cartesian closed, we can use its diagonal. But we need to also construct a map of shape $U \times X \times X \to X$, so it would be nice to have a monoid structure in X. It turns out that simply asking for free commutative monoids in $\mathbb{C}$ is enough.

This corresponds to the fact that our category $Dial_2(\mathbb{C})$ has objects with dual aspects to it. The left-handed object U, in the triple (U, X, α), tends to behave as expected in a cartesian closed category, that is, it works like a verifier of assertions. Meanwhile, the right-handed side object, X, tends to behave like a producer of counter-examples to the original claim made by the whole object $A = (U, X, \alpha)$ that, internally, there exists an u in U, such that for all x in X, $\alpha(u, x)$ holds.

Thus in our case, to obtain a comonoid in $Dial_2(\mathbb{C})$ we simply need an object that is a monoid in the second coordinate, this not only defines a comonad, but this comonad is a cofree one.

▶ **Definition 3.** *Define a comonad in $Dial_2(\mathbb{C})$ by saying $!(U, X, \alpha) = (U, X^*, \alpha^*)$, where X^* is the free commutative monoid in $\mathbb{C}$ and the new relation α^* is simply a repetition of the α relation, as many times as necessary for the sequences in X^*.*

Then we prove that this is the cofree comonad in $Dial_2(\mathbb{C})$, that is, that there is an adjunction between the forgetful functor from the category of commutative comonoids in $Dial_2(\mathbb{C})$ and $Dial_2(\mathbb{C})$ itself.

▶ **Theorem 4** (de Paiva 1987). *The monoidal comonad ! in $Dial_2(\mathbb{C})$ models linear logic modalities and recovers intuitionistic propositional logic.*

To prove this, we transform it into

▶ **Lemma 5.** *The co-Kleisli category associated with the (symmetric monoidal) comonad ! on $Dial_2(\mathbb{C})$ is cartesian closed.*

Proof. To show cartesian closedness we need to show:

$$Hom_{Kl!}(A\&B, C) \cong Hom_{Kl!}(A, [B, C]_{Kl!})$$

The proof is then a series of equivalences that were proved previously:
$Hom_{Kl!}(A\&B, C) \cong Hom_{Dial_2(\mathbb{C})}(!(A\&B), C) \cong Hom_{Dial_2(\mathbb{C})}(!A \otimes !B, C)$
$\cong Hom_{Dial_2(\mathbb{C})}(!A, [!B, C]_{Dial_2(\mathbb{C})}) \cong Hom_{Kl!}(A, [!B, C]_{Dial_2(\mathbb{C})}) \cong Hom_{Kl!}(A, [B, C]_{Kl!})$
This is described with details in [28] and section 2.5 of [14]. ◀

This was the first cofree comonad model of linear logic, not purely syntactic, that is, not simply of the form this is what we want to be the case. It has inspired the comonad in Chu spaces, as described by Lafont and Streicher [22].

The operator ! is a monoidal comonad, however, the significance of *monoidal* was not realized to begin with. That means that there is a natural transformation

$$m_{(-,-)} : !A \otimes !B \to !(A \otimes B) \text{ and a morphism } M_I : I \to !I$$

satisfying many commutative diagrams. The operator ! induces a commutative comonoid structure on the object $!A$, but $!A$ also has naturally a coalgebra structure induced by the comonad !. The comonoid and coalgebra structures must interact in a nice way and how to describe this nice way is up to controversy.

4.2 Exponential comonads for Dialectica-like constructions

The Dialectica-like category $DDial_2(\mathbb{C})$ [15] is a simpler category than $Dial_2(\mathbb{C})$. Since we simplified the category, does it mean we simplified the exponential comonad used to embed it into a traditional cartesian universe? Unfortunately, the answer is no. The previous comonad we had for $!A$ in $Dial_2(\mathbb{C})$ does not work for $DDial_2(\mathbb{C})$: we need commutative comonoids and coalgebras with respect to the new tensor product, which is much more complicated than the tensor product of $Dial_2(\mathbb{C})$.

Tensor products should be adjoint to internal-homs, which internalize the morphisms of the category under discussion. The category $DDial_2(\mathbb{C})$ has simpler morphisms than $Dial_2(\mathbb{C})$: the internal-hom of $DDial_2(\mathbb{C})$ is given by

$$A \multimap B \quad = \quad (V^U \times X^Y \xleftarrow{\alpha \multimap \beta} U \times Y)$$

where counterexample morphisms are simply from Y to X, as explained before. We can reverse engineer a tensor product that works well with the simplified internal-hom of $DDial_2(\mathbb{C})$ above. This tensor product is given by

$$(U, X, \alpha) \otimes (V, Y, \beta) = (U \times V, X^V \times Y^U, \alpha \otimes \beta).$$

But this is more complicated than simply putting relations side-by-side as we had for the category $Dial_2(\mathbb{C})$.

Given that the the previous comonad ! worked so well in the category $Dial_2(\mathbb{C})$ and that it is still a comonad in $DDial_2(\mathbb{C})$, we can call it $\mathtt{bang}_1$ and use it to deal with the comonoid structure. Note that in the previous definition we never used the dependency in U of the morphisms considered.

▶ **Definition 6.** *Define a functor in $DDial_2(\mathbb{C})$ by saying $\mathtt{bang}_1(U, X, \alpha) = (U, X^*, \alpha^*)$, where X^* is the free commutative monoid in $\mathbb{C}$ and the new relation α^* is simply a repetition of the α relation, as many times as necessary for the sequences in X^*.*

To have coalgebras we need a different comonad.

▶ **Definition 7.** *Define a functor in $DDial_2(\mathbb{C})$ by saying $\mathtt{bang}_2(U, X, \alpha) = (U, X^U, \alpha^U)$, where $(-)^U : \mathbb{C} \to \mathbb{C}$ is the corresponding monad, for a given U, in $\mathbb{C}$.*

This induces a comonad in $\mathsf{DDial}_2(C)$ but $\mathtt{bang}_2$ isn't enough!

The good news is that we can compose these comonads, unlike most comonads. We just need a distributive law, which had been already described by Beck [3]. (The suggestion of using a composite of comonads came from Th. Coquand, and it was much appreciated.)

Take $\mathtt{bang}(U, X, \alpha) = (U, (X^*)^U, (\alpha*)^U)$, where $(-)^*$ is the free commutative monoid in $\mathbb{C}$ and $(-)^U : \mathbb{C} \to \mathbb{C}$ is a monad in $\mathbb{C}$ that induces a comonad in $\mathsf{DDial}_2(C)$. Most of the calculations are in [14] and [15], except the ones for monoidicity of the (co)monads involved.

▶ **Proposition 8.** *There is a comonad* bang *in $\mathsf{DDial}_2(\mathbb{C})$, a composite of two comonads, which models the modality ! in Linear Logic.*

▶ **Theorem 9** (Linear-Non-Linear version)**.** *There is a* **monoidal** *adjunction between $\mathsf{DDial}_2(\mathbb{C})$ and its coKleisli category for the composite monoidal comonad* bang *described above.*

A comonad that is a composite of two comonads is harder to deal with than the situation we had with $\mathsf{Dial}_2(\mathbb{C})$. The fact that this comonad is obtained from a composite monad in $\mathbb{C}$ and that we need to use two distributive laws (one for the monads in $\mathbb{C}$, one for the comonads in $\mathsf{DDial}_2(\mathbb{C})$) makes the whole enterprise somewhat convoluted. One might be led to think that this is a "hack", without further significance. The calculations get bigger and cumbersome, but do not present intrinsic difficulties. They require a small amount of Street's theory of monads [31] suitably dualized in places.

But something more interesting is happening here: the comonads come with their own appropriate tensor products, as they need to be monoidal, for logical reasons. The comonad ! in $\mathsf{Dial}_2(\mathbb{C})$ is very well-behaved, it is cofree. The composite comonad bang in $\mathsf{DDial}_2(\mathbb{C})$ seems very *ad hoc*, it is clearly not cofree, but it is monoidal and this is enough for the modelling role we wanted it for. The calculations showing that all the commutative diagrams really commute are easy, but there are many of them and they are long. Computer systems can help with them, e.g. the Agda report from Eades in `https://github.com/heades/cut-fill-agda` for the paper [16].

These constructions look ad hoc, but they have been used repeatedly and reappear in many other projects, starting with [8] and [11]. We describe this use in the next section. We also draw the reader's attention to the formal similarity of these notions of multiple tensors with associated comonads to the work of Retoré in [26] and of Laird anf Churchill on sequentiality [7].

5 Modelling State?

The concept of state-manipulation plays an important role in computation. There are many suggestions of how to formalise this concept, especially using type systems derived from Girard's Linear Logic. Linear Logic was long considered a candidate logic for modelling "state" as usually considered in programming languages. Reddy introduced his system LLMS (Linear Logic Model of State) in 1993 [25] to address the issue of state in functional and imperative programming languages. As he says "This work addresses twin issues: how to incorporate state manipulation in functional programming languages, and how to describe the semantics of higher-order imperative programming languages."

Reddy's system is an extended intuitionistic linear calculus with extra connectives for modelling state manipulation. The system LLMS adds two extra connectives to those of Intuitionistic Linear Logic: first, a sequencing binary operator $\triangleright$, called "before", which is a non-commutative tensor product and a "regenerative" storage operator $\dagger$, a modality associated with "before", in a way that parallels the relationship between tensor and the exponential !.

Discussing the system, Reddy explains the connective 'before' $\triangleright$ as the denotation of sequential composition of components. Meanwhile the "regenerative" storage operator $\dagger$ allows him to build sequentially reusable storage objects. The approach is shown to work by embbeding a higher-order Algol-like language in the system LLMS.

If "before" captures composition with possible left to right communication, it stands to reason that it should be non-commutative. But we can consider a commutative variant of the connective, obtaining a non-ordered combination of elements. This was the work in [8]. In that work non-ordered combination does not mean communication both ways. Instead it represents concurrent execution without interaction or synchronization, which was a suggestion of Haeusler.

The work with Correa and Haeusler [8] follows exactly Reddy's description, which in its turn is based on Retoré Pomset logic [26]. Pomset logic is not an easy system, and it is particularly difficult, if "proof-nets" are not your area of expertise. A context Γ is characterized as a pomset (a partially ordered multiset) $(|\Gamma|, \leq_\Gamma)$, where $|\Gamma|$ is the set of formula occurrences in Γ, and $\leq_\Gamma$ is a partial order on the context $|\Gamma|$. It turns out that the commutative version of LLMS, LLMS_c, can be soundly modeled by the dialectica category $\mathsf{DDial}_2(\mathbb{C})$ and LLMS itself can be modelled by a suitable (non-commutative) variant of the dialectica construction [11]. This non-commutative variant of the dialectica construction had first been considered as a model for the Lambek calculus in [10].

The intrinsic order of the pomset context makes proofs much harder and leads into issues discussed by the community interested in Deep Inference. Many questions are still open, as discussed recently by Retoré [27] and Slavnov [29]. But some answers can be provided.

▶ **Theorem 10** (cut elimination). *If a sequent is provable in the commutative system $LLMS_c$, then it is provable in $LLMS_c$ without any application of the cut rule.*

Cut elimination together with the theorem below, are the main results of [8], together with the interpretation in terms of CSP processes of formulae of the system.

▶ **Theorem 11** (Corrêa, Haeusler, de Paiva, 1996). *The dialectica categories $\mathsf{DDial}_2(\mathbb{C})$ are a sound model of $LLMS_c$.*

The fact that the ad hoc construction of the dialectica-like modality was recreated, independently, in the totally unrelated field of managing state in functional and imperative programming languages, makes one wonder. The construction seems to be robust. As it is shown in [11] we also have

▶ **Theorem 12** (Tucker 1998). *There exists a small variant of the dialectica categories* $\mathsf{DDial}_2(\mathbb{C})$*, where the original, non-commutative system LLMS can be soundly modeled.*

6 Conclusions

We discussed some stories of monads, comonads, algebras and coalgebras that appear in work related to logic systems oriented towards computing. My main examples have to do with Linear Logic, so I called these "Linear Modalities" as (co)monads behave like S4 modalities when considered as components of proof systems over a linear basis. I hope every one can see how Dialectica spaces of several kinds introduce different (co)monads useful to provide models of several possible extensions of linear logic.

I have not discussed much other constructive modal logics, which is also work that goes under the heading of (co)monads and (co)algebras for Computer Science. Constructive modal logics are interesting for programmers, logicians, mathematicians and philosophers who know about them under different names and for different applications. It is a shame that these communities do not talk much to each other. The consortium of researchers united under the title "Intuitionistic Modal Logics and Applications" (IMLA) [30, 13, 12] has been trying to change this since 1999. It would be good if this work could be continued and improved. There is plenty of future work along these lines, producing categorical models for systems with linear modalities of different styles.

I have started the talk, that is now written as this note, with a quote from Martin Hyland in "Proof Theory in the Abstract" [18]. He says

> Elegant mathematics will of itself tell a tale, and one with the merit of simplicity. This may carry philosophical weight. But that cannot be guaranteed: in the end one cannot escape the need to form a judgement of significance.

Working within the confines of categorical versions of the Curry-Howard correspondence has been a guiding principle for me, others can find their own judgements of significance. There is plenty of interesting work to do.

References

1. Paul André Melliès. Categorical semantics of linear logic. In *In: Interactive Models of Computation and Program Behaviour, Panoramas et Synthèses 27, Société Mathématique de France 1–196*, 2009.
2. Andrew Barber and Gordon Plotkin. *Dual intuitionistic linear logic.* University of Edinburgh, Department of Computer Science, Laboratory for . . . , 1996.
3. Jon Beck. Distributive laws. In B. Eckmann, editor, *Seminar on Triples and Categorical Homology Theory*, pages 119–140, Berlin, Heidelberg, 1969. Springer Berlin Heidelberg.
4. Nick Benton, Gavin Bierman, Valeria De Paiva, and Martin Hyland. A term calculus for intuitionistic linear logic. In *International Conference on Typed Lambda Calculi and Applications*, pages 75–90. Springer, 1993.
5. P. N. Benton. A mixed linear and non-linear logic: Proofs, terms and models. In Leszek Pacholski and Jerzy Tiuryn, editors, *Computer Science Logic*, pages 121–135, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.
6. Gavin M Bierman. What is a categorical model of intuitionistic linear logic? In *International Conference on Typed Lambda Calculi and Applications*, pages 78–93. Springer, 1995.
7. Martin Churchill and James Laird. A logic of sequentiality. In *International Workshop on Computer Science Logic*, pages 215–229. Springer, 2010.

8 Marcelo da S Corrêa, Edward H Haeusler, and Valeria CV de Paiva. A dialectica model of state. In *CATS'96, Computing: The Australian Theory Symposium Proceedings*, pages 11–12, 1996.

9 Valeria de Paiva. The dialectica categories. *Categories in Computer Science and Logic*, 92:47–62, 1989.

10 Valeria De Paiva. A dialectica model of the lambek calculus. In *AMSTERDAM COLLOQUIUM, 1991*. Citeseer, 1991.

11 Valeria De Paiva. Linear logic model of state revisited. *Logic Journal of the IGPL*, 22(5):791–804, 2014.

12 Valeria de Paiva, Rajeev Goré, and Michael Mendler. Modalities in constructive logics and type theories, 2004.

13 Valeria de Paiva and Brigitte Pientka. Intuitionistic modal logic and applications (IMLA 2008). *Information and computation (Print)*, 209(12), 2011.

14 Valeria Correa Vaz de Paiva. The dialectica categories. Technical report, University of Cambridge, Computer Laboratory, Technical Report 213, 1991. URL: `https://128.232.0.20/techreports/UCAM-CL-TR-213.pdf`.

15 Valeria CV De Paiva. A dialectica-like model of linear logic. In *Category Theory and Computer Science*, pages 341–356. Springer, 1989.

16 Harley Eades and Valeria de Paiva. Multiple conclusion linear logic: Cut elimination and more. In Sergei Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science*, pages 90–105, Cham, 2016. Springer International Publishing.

17 Jean-Yves Girard and Yves Lafont. Linear logic and lazy computation. In *International Joint Conference on Theory and Practice of Software Development*, pages 52–66. Springer, 1987.

18 J.M.E. Hyland. Proof theory in the abstract. *Annals of Pure and Applied Logic*, 114(1):43–78, 2002. Troelstra Festschrift. `doi:10.1016/S0168-0072(01)00075-6`.

19 Martin Hyland and Valeria De Paiva. Full intuitionistic linear logic. *Annals of Pure and Applied Logic*, 64(3):273–291, 1993.

20 Martin Hyland and Andrea Schalk. Glueing and orthogonality for models of linear logic. *Theoretical computer science*, 294(1-2):183–231, 2003.

21 Ulrich Kohlenbach. *Applied proof theory: proof interpretations and their use in mathematics*. Springer Science & Business Media, 2008.

22 Yves Lafont and Thomas Streicher. Games semantics for linear logic. In *Proceedings 1991 Sixth Annual IEEE Symposium on Logic in Computer Science*, pages 43–44. IEEE Computer Society, 1991.

23 Maria Emilia Maietti, Paola Maneggia, Valeria De Paiva, and Eike Ritter. Relating categorical semantics for intuitionistic linear logic. *Applied categorical structures*, 13(1):1–36, 2005.

24 Vaughan Pratt. Chu spaces. *School on Category Theory and Applications (Coimbra, 1999)*, 21:39–100, 1999.

25 Uday S. Reddy. A linear logic model of state. *Manuscript*, 1993.

26 Christian Retoré. Pomset logic: A non-commutative extension of classical linear logic. In Philippe de Groote and J. Roger Hindley, editors, *Typed Lambda Calculi and Applications*, pages 300–318, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

27 Christian Retoré. Pomset logic: a logical and grammatical alternative to the lambek calculus, 2020. `arXiv:2001.02155`.

28 R.A.G. Seely. Linear logic, *-autonomous categories and cofree coalgebras. In *Categories in Computer Science and Logic*, pages 371–382. American Mathematical Society, 1989.

29 Sergey Slavnov. On noncommutative extensions of linear logic. *arXiv preprint arXiv:1703.10092*, 2017.

30 Charles Stewart, Valeria de Paiva, and Natasha Alechina. Intuitionistic modal logic: a 15-year retrospective. *Journal of Logic and Computation*, 2018.

31 Ross Street. The formal theory of monads. *Journal of Pure and Applied Algebra*, 2(2):149–168, 1972. `doi:10.1016/0022-4049(72)90019-9`.