

## Linear logic model of state revisited\*

Valeria de Paiva

Rearden Commerce,  
Foster City, CA 94405, USA  
Valeria.dePaiva@gmail.com

**Abstract.** In an unpublished note, Reddy introduced an extended intuitionistic linear calculus, called LLMS (for Linear Logic Model of State), to model state manipulation via the notions of sequential composition and ‘regenerative values’. His calculus introduces the connective “before”  $\triangleright$  and an associated modality  $\dagger$ , for the storage of objects sequentially reusable. Meanwhile independently de Paiva introduced a (collection of) dialectica categorical models for (classical and intuitionistic) Linear Logic, the categories  $\text{Dial}_2\text{Set}$ . These categories contain, apart from the structure needed to model linear logic, an extra tensor product functor  $\otimes$  and a comonad structure corresponding to a modality related to this tensor.

It is surprising that these works arising from completely different motivations can be related in a meaningful way. But in this paper, following our joint work with Corrêa and Haeusler, we first adapt Reddy’s system LLMS providing it with a commutative version of the connective “before”, denoted by  $\odot$ , and an associated modality and construct a dialectica category on **Sets**, which we show is a sound model for the modified system  $\text{LLMS}_c$ . Moreover, following the work of Tucker, we provide another variant of the Dialectica categories with a non-commutative tensor and associated modality, which models soundly LLMS itself. We conclude with some speculation on future applications.

*Keywords* Intuitionistic Linear Logic, Sequentiality, Models of State, Proof Theory, Categorical Models.

### Preamble

This paper is heavily based on “A dialectica model of state” by Marcelo da Silva Corrêa, E. Hermann Haeusler and myself, presented by Marcelo (then Hermann’s PhD student) at CATS: Computing, the Australian Theory Symposium, in January 1996 [5]. The paper has never been published in a journal, it only exists in the informal proceedings of that conference, so it is hard to find and hard to cite. But work on modelling state manipulation using linear logic ideas is still flourishing. Mostly via the connection to game semantics and the work of G. McCusker, P. O’Hearn and U. Reddy himself, many ideas are still being being debated and improved upon. Thus the work on linear logic model of state seems to me an ideal contribution to this tribute to E. Hermann Haeusler.

\* I would like to thank Marcelo da Silva Corrêa not only for the work we did together many years ago, but also for keeping the files that allowed me to produce this version to celebrate E. Hermann Haeusler’s birthday.

It is a real pleasure for me to acknowledge the extent of my collaboration, over the years, with Hermann. Through the ups and downs of the funding situations in Brazil, England and the United States of America, we have managed to work together in several projects, some more official, some more personal, all, always, a lot of fun. Hermann has always been extremely generous with his time and that of his students and through him I have had the privilege of working with many young talented people. Celebrating Hermann’s birthday seems to me the best way of bringing the discussion on imperative notions of state and linear logic back into focus. Happy Birthday Hermann!

## 1 Introduction

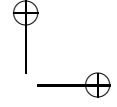
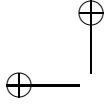
The concept of state-manipulation plays an important role in computation. There are many attempts to formalise this concept especially using type systems derived from Girard’s Linear Logic [9, 11]. Some aspects of state-manipulation, like sequencing and regenerative values, are captured in an interesting way by Reddy’s work on the typing system described in [11]. Reddy presents an extended intuitionistic linear calculus for modelling state manipulation; his calculus LLMS (for Linear Logic Model of State) adds two extra connectives to those of Intuitionistic Linear Logic: a sequencing binary operator  $\triangleright$ , called “before”, and a “regenerative” storage operator  $\dagger$ , a modality associated with “before”, in a way that parallels the relationship between tensor and the exponential !.

The connective “before” is motivated as the denotation of sequential composition of components while the “regenerative” storage operator allows us to build sequentially reusable storage objects. These two constructors are used to express dynamic values and imperative programs within the framework of linear logic, while Girard’s “of course” modality is used to express static values. The approach is shown to work by embedding a higher-order Algol-like language in the system LLMS.

Reddy’s intuitive idea is that the connective “before” captures the computational feature of composition with possible one way (left to right) communication. Thus this connective must be noncommutative. If one considers a variant of this connective which is commutative, one obtains an operator denoting non-ordered (or aleatory) combination of elements. Non-ordered combination does not mean here communication both ways, but it can be said to represent concurrent execution without interaction or synchronization. In this paper we first introduce a new connective  $\odot$  which is just the commutative version of Reddy’s  $\triangleright$  and we modify the system LLMS accordingly, to provide a presentation of a system  $LLMS_c$ , for commutative LLMS.

Reddy presents a specific categorical model for his calculus using a variant of coherent spaces and linear maps. However, considering the commutative version of the connective “before”, related to interleaving as we suggest in [5], we end up with the calculus  $LLMS_c$  for which we had already a dialectica categorical model many years before.

Dialectica models were introduced by de Paiva [6] as a model for the intuitionistic fragment of Linear Logic, that arises from Gödel’s Dialectica Interpretation, hence the name. Later, de Paiva introduced the categories  $GC$  [7, 8], a simplified kind of dialectica model, which following a suggestion of Girard, model the whole of Linear Logic.



The categories  $GC$  have been re-christened  $\text{Dial}_2(\text{Set})$ . The categories  $\text{Dial}_2(\text{Set})$  have exactly the extra structure (an extra tensor product and an extra modality, associated with this tensor product) to model  $\text{LLMS}_c$ . The extra tensor product, denoted here by  $\odot$ , can be seen as representing an aleatory combination of elements. The extra (monoidal) comonad can be seen as a commutative version of the “regenerative” storage operator.

We first (section 2) describe the proof system  $\text{LLMS}_c$ , obtained by modifying  $\text{LLMS}$  to make the connective “before” commutative. In the section 3, we describe the specific dialectica  $\mathcal{G}$  category (built over  $\text{Sets}$ ) and show that it is a sound model for  $\text{LLMS}_c$ . Then we recap the work of Tucker [14] where a variation of the construction produces a model of  $\text{LLMS}$  itself. Finally, we conclude pointing out some future work.

## 2 The system $\text{LLMS}_c$

The presentation of the system of  $\text{LLMS}_c$  follows strictly the presentation of  $\text{LLMS}$  [11], which itself follows from Retoré’s work [13] on *pomset logic*.

The left context of a sequent in  $\text{LLMS}_c$  has the following syntax :

$$\Gamma ::= \varepsilon | A | \Gamma_0, \Gamma_1 | \Gamma_0; \Gamma_1$$

where  $\varepsilon$  (empty context) is interpreted as  $\mathbf{1}$  and the contexts  $\Gamma_0, \Gamma_1$  and  $\Gamma_0; \Gamma_1$  are interpreted as  $\Gamma_0 \otimes \Gamma_1$  and  $\Gamma_0 \odot \Gamma_1$ , respectively. A context  $\Gamma$ , which does not contain any occurrences of the connective “;”, is called an *independent context*.

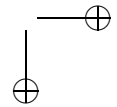
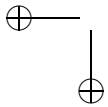
A context  $\Gamma$  is characterized as a pomset (partially ordered multiset)  $(|\Gamma|, \leq_\Gamma)$ , where  $|\Gamma|$  is the set of formula occurrences in  $\Gamma$ , and  $\leq_\Gamma$  is a partial order on  $|\Gamma|$ . They are defined inductively as follows:

$$\begin{aligned} |\varepsilon| &= \{\} \\ |A| &= \{A\} \quad \leq_A = \{(A, A)\} \\ |\Gamma_1, \Gamma_2| &= |\Gamma_1| + |\Gamma_2| \quad \leq_{\Gamma_1, \Gamma_2} = \leq_{\Gamma_1} \cup \leq_{\Gamma_2} \\ |\Gamma_1; \Gamma_2| &= |\Gamma_1| + |\Gamma_2| \quad \leq_{\Gamma_1; \Gamma_2} = \leq_{\Gamma_1} \cup \leq_{\Gamma_2} \cup (|\Gamma_1| \times |\Gamma_2|) \cup (|\Gamma_2| \times |\Gamma_1|) \end{aligned}$$

The rules of  $\text{LLMS}_c$ , are just those of  $\text{LLMS}$  [11], except that we use the symbol  $\odot$  instead of  $\triangleright$ . The fact that  $\odot$  is commutative, whereas  $\triangleright$  is not, is “hidden” in the definition of the contexts. The rules of  $\text{LLMS}_c$  are presented in the Figure 1.

Figure 1 also works as a representation of the rules of  $\text{LLMS}$ , if we replace in the definition of contexts the last rule by:

$$|\Gamma_1; \Gamma_2| = |\Gamma_1| + |\Gamma_2| \text{ where } \leq_{\Gamma_1; \Gamma_2} = \leq_{\Gamma_1} \cup \leq_{\Gamma_2} \cup (|\Gamma_1| \times |\Gamma_2|)$$



*Structural Rule*

$$\frac{\Gamma' \vdash A}{\Gamma \vdash A} \mathbf{Ser} \quad \text{if } |\Gamma| = |\Gamma'| \text{ and } \leq_{\Gamma} \subseteq \leq_{\Gamma'}$$

*Identity Rules*

$$\frac{}{A \vdash A} \mathbf{Id} \quad \frac{\Gamma \vdash A \quad \Delta[A] \vdash B}{\Delta[\Gamma] \vdash B} \mathbf{Cut}$$

*Multiplicative Rules*

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} \otimes \mathcal{R} \quad \frac{\Gamma[A, B] \vdash C}{\Gamma[A \otimes B] \vdash C} \otimes \mathcal{L} \quad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma; \Delta \vdash A \odot B} \odot \mathcal{R} \quad \frac{\Gamma[A; B] \vdash C}{\Gamma[A \odot B] \vdash C} \odot \mathcal{L}$$

$$\frac{}{\vdash \mathbf{1}} \mathbf{1R} \quad \frac{\Gamma[\varepsilon] \vdash C}{\Gamma[\mathbf{1}] \vdash C} \mathbf{1L} \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B} \multimap \mathcal{R} \quad \frac{\Gamma \vdash A \quad \Delta[B] \vdash C}{\Delta[\Gamma, A \multimap B] \vdash C} \multimap \mathcal{L}$$

*Additive Rules*

$$\frac{}{\vdash \top} \top \mathcal{R} \quad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \& B} \& \mathcal{R} \quad \frac{\Gamma[A] \vdash C}{\Gamma[A \& B] \vdash C} \mathbf{1} \& \mathcal{L} \quad \frac{\Gamma[B] \vdash C}{\Gamma[A \& B] \vdash C} \mathbf{2} \& \mathcal{L}$$

*Modalities*

$$\frac{\Gamma[\varepsilon] \vdash C}{\Gamma[\dagger A] \vdash C} \dagger \mathbf{Weak} \quad \frac{\Gamma[A] \vdash C}{\Gamma[\dagger A] \vdash C} \dagger \mathbf{Der} \quad \frac{\Gamma[\dagger A; \dagger A] \vdash C}{\Gamma[\dagger A] \vdash C} \dagger \mathbf{Thread}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash \dagger A} \quad (\text{if } \Gamma \text{ is an independent context with only } ! \text{ or } \dagger \text{ formulae})$$

$$\frac{\Gamma[\dagger A] \vdash C}{\Gamma[!A] \vdash C} \mathbf{!Ser} \quad \frac{\Gamma[!A, !A] \vdash C}{\Gamma[!A] \vdash C} \mathbf{!Contr}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash !A} \quad (\text{if } \Gamma \text{ is an independent context with only } ! \text{ formulae})$$

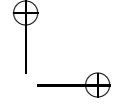
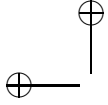
Figure 1: Proof system of LLMS<sub>c</sub>.

## 2.1 Interplay between tensor structures

Next, we describe some inferences which are derived from the structural rule and which show some interesting interplay between the connectives  $\otimes$  and  $\odot$ .

**Proposition 1.** *The following inferences are allowed by the Structural Rule (Ser).*

$$\begin{aligned} \text{(i)} \quad & \frac{\Delta[(A, B); C] \vdash D}{\Delta[A, (B; C)] \vdash D} & \text{(ii)} \quad & \frac{\Delta[A; (B, C)] \vdash D}{\Delta[(A; B), C] \vdash D} \\ \text{(iii)} \quad & \frac{\Delta[A; B] \vdash C}{\Delta[A, B] \vdash C} & \text{(iv)} \quad & \frac{\Delta[(A, C); (B, D)] \vdash E}{\Delta[(A; B), (C; D)] \vdash E} \end{aligned}$$



Commutativity and associativity of the tensor products are also consequences of the structural rule. The *weak-distributivity* properties

$$\begin{aligned} A \otimes (B \odot C) &\multimap (A \otimes B) \odot C \\ (A \odot B) \otimes C &\multimap A \odot (B \otimes C) \end{aligned}$$

follow from the inferences (i) and (ii), respectively. The inferences (iii) and (iv) allow us to prove that

$$\begin{aligned} A \otimes B &\multimap A \odot B \\ (A \odot B) \otimes (C \odot D) &\multimap (A \otimes C) \odot (B \otimes D) \end{aligned}$$

Summing up and naming the transformations, the following are direct consequences of the structural rule Ser:

$$\begin{aligned} \text{ser} &: A \otimes B \rightarrow A \odot B \\ \text{wd1} &: A \otimes (B \odot C) \rightarrow (A \otimes B) \odot C \\ \text{wd2} &: (A \odot B) \otimes C \rightarrow A \odot (B \otimes C) \\ \text{int} &: (A \odot B) \otimes (C \odot D) \rightarrow (A \odot C) \otimes (B \odot D) \end{aligned}$$

They all draw on the fact that we can impose the same order on formulae as we can on contexts, by turning  $\otimes$ 's into commas,  $\odot$ 's into semicolons and applying  $\leq$  to the resulting context. A formula on the left can then be transformed into the one on the right if the order on the left formula is included in the order on the right formula.

## 2.2 Cut Elimination for LLMS

We can obtain a Gentzen's style proof of the cut-elimination theorem for  $\text{LLMS}_c$ .

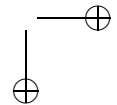
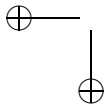
**Theorem 1 (Cut Elimination).** *If a sequent is provable in  $\text{LLMS}_c$ , then it is provable in  $\text{LLMS}_c$  without an application of the cut rule.*

This proof is probably similar to Reddy's proof [11], but we consider as a measure of the complexity, the rank as well as the degree of the last cut-formula, with the usual definitions. Special attention must be paid to the cases involving  $\dagger$ , for instance, the case in which the  $\dagger$ **Thread** is applied in the lower right sequent of a proof ending by a cut rule is proved using the inferences (iii) and (iv). Corrêa's proof can be found in the technical report [4].

## 3 A Categorical Model for $\text{LLMS}_c$

To characterize  $\text{LLMS}_c$  categorically, we consider an instantiation  $\mathcal{G}$  of the (symmetric monoidal closed) categories  $\text{Dial}_2(\text{Set})$ , developed in [8]. As usual, the interpretation of the formulae is given by objects of the category  $\mathcal{G}$  and proofs of  $\text{LLMS}_c$  are interpreted by morphisms of  $\mathcal{G}$ .

Recall that the partial-order with two elements  $\mathbf{2}$ , where one thinks of  $\mathbf{0}$  as false and  $\mathbf{1}$  as true, is closed as a poset. It has a tensor product ( $\otimes$ ) and an internal-hom ( $\multimap$ ) such that  $a \otimes b \leq c$  iff  $a \leq b \multimap c$ , where  $\otimes$  is the usual conjunction  $\wedge$  and the internal-hom *linear* is the usual Heyting implication  $\rightarrow$ .



**Definition 1 (dialectica category  $\mathcal{G}$ ).** The category  $\mathcal{G}$  is described by:

- Objects are relations on **Sets**, that is maps  $U \times X \xrightarrow{\alpha} \mathbf{2}$ , written as  $(U \xleftarrow{\alpha} X)$ .
- A morphism from  $(U \xleftarrow{\alpha} X)$  to  $(V \xleftarrow{\beta} Y)$  consists a pair of maps,  $f: U \rightarrow V$  and  $F: Y \rightarrow X$ , such that the following condition is satisfied

$$\alpha(u, Fy) \leq \beta(fu, y)$$

where  $\leq$  is the usual order in  $\mathbf{2}$ .

Thus there is a morphism  $(f, F)$  from  $\alpha$  to  $\beta$  iff for all  $u$  in  $U$  and all  $y$  in  $Y$ , if  $u\alpha F(y)$  then  $f(u)\beta y$ . We usually depict a morphism  $(f, F)$  in  $\mathcal{G}$  as follows, to help remember what happens in which coordinate:

$$\begin{array}{ccc} U & \xleftarrow{\alpha} & X \\ f \downarrow & & \uparrow F \\ V & \xleftarrow{\beta} & Y \end{array}$$

The next definition describes the (multiplicative and additive) structure of the category  $\mathcal{G}$  we are interested in.

**Definition 2 (structure of  $\mathcal{G}$ ).** Let  $A = (U \xleftarrow{\alpha} X)$  and  $B = (V \xleftarrow{\beta} Y)$  be objects of the category  $\mathcal{G}$ .

- The tensor bifunctor  $\otimes: \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}$  is given by

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

where  $(u, v)\alpha \otimes \beta(f, g)$  iff  $u\alpha f(v)$  and  $v\beta g(u)$ . Its unit is the object  $I = (\mathbf{1} \xleftarrow{\iota} \mathbf{1})$ , where  $\iota$  is the identity relation on  $\mathbf{1}$ .

- The tensor bifunctor  $\odot: \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}$  is given by

$$A \odot B = (U \times V \xleftarrow{\alpha \odot \beta} X \times Y),$$

where  $(u, v)\alpha \odot \beta(x, y)$  iff  $u\alpha x$  and  $v\beta y$ . Its unit is the same object  $I$  above.

- The internal-hom bifunctor  $[-, -]_{\mathcal{G}}: \mathcal{G}^{op} \times \mathcal{G} \rightarrow \mathcal{G}$  is given by

$$[A, B]_{\mathcal{G}} = (V^U \times X^Y \xleftarrow{\beta^\alpha} U \times Y),$$

where  $(f, F)\beta^\alpha(u, y)$  iff  $u\alpha x$  implies  $v\beta y$ . The relation  $\beta^\alpha$  is given by the composition

$$V^U \times X^Y \times U \times Y \xrightarrow{(\pi_3, \pi_4, \text{eval}, \text{eval})} U \times Y \times V \times X \xrightarrow{\alpha \times \beta'} \mathbf{2} \times \mathbf{2} \xrightarrow{-\circ} \mathbf{2}.$$

- The product bifunctor  $\&: \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}$  is given by

$$A \& B = (U \times V \xleftarrow{\alpha \& \beta} X + Y),$$

where  $(u, v) \alpha \& \beta \begin{pmatrix} x, 0 \\ y, 1 \end{pmatrix}$  iff either  $u \alpha x$  or  $v \beta y$ . Its unit is the object  $\mathbf{T} = (\mathbf{1} \leftarrow \mathbf{0})$ .

We could also describe coproducts (or additive disjunctions), a par connective and the linear negation, but these will not play an important role in the following discussion.

**Lemma 1** ( $\mathcal{G}$  is a smcc). *With the structure just described the category  $\mathcal{G}$  is a symmetric monoidal closed category.*

The proof is easy, but worth doing on your own, to appreciate the naturalness of the categorical constructions.

**Lemma 2** ( $\mathcal{G}$  natural transformations). *For any  $A$  and  $B$  objects of  $\mathcal{G}$  there are natural morphisms:*

- (i)  $\text{ser}_{(A,B)} : A \otimes B \rightarrow A \odot B$ ; <sup>1</sup>
- (ii)  $\text{wd1}_{(A,B,C)} : A \otimes (B \odot C) \rightarrow (A \otimes B) \odot C$  (called weak-distributivity 1)
- (iii)  $\text{wd2}_{(A,B,C)} : (A \odot B) \otimes C \rightarrow A \odot (B \otimes C)$  (called weak-distributivity 2)
- (iv)  $\text{int}_{(A,B,C,D)} : (A \odot B) \otimes (C \odot D) \rightarrow (A \otimes C) \odot (B \otimes D)$ .

Following the pattern of linear logic, modelling the modalities of  $\text{LLMS}_c$ , is more complicated than the tensors. Recall that the category **Sets** does have free (commutative) monoid structures. There is an adjunction

$$F \dashv U : \mathbf{Mon}_c \rightarrow \mathbf{Sets}$$

where  $\mathbf{Mon}_c$  is the category of commutative monoids  $U$  is the forgetful functor and the free functor  $F$  applied to a set  $X$

$$F(X) = (X^*, e_X, m_X)$$

consists of  $X^*$  the set of finite sequences (up to permutation) of elements of  $X$ ,  $e_X : \mathbf{1} \rightarrow X^*$  is the empty sequence and  $m_X : X^* \times X^* \rightarrow X^*$  means concatenation of sequences.

We need to consider another, different, monad structure in **Sets** too. Consider the monad  $(T_U, \eta_{T_U}, \mu_{T_U})$ , where for a given  $U$  in **Sets**,  $T_U : \mathbf{Sets} \rightarrow \mathbf{Sets}$  is the endofunctor which takes  $X \mapsto X^U$ ,  $Y \mapsto Y^U$  and if  $g : X \rightarrow Y$ ,  $f \in X^U$  is taken to  $f; g \in Y^U$ . Its unit is given by the transpose of the projection, that is the constant map  $(\eta_{T_U})_X : X \rightarrow X^U$  and its multiplication  $(\mu_{T_U})_X : X^{U \times U} \rightarrow X^U$ , is simply precomposition with the diagonal map on  $U$ .

We define two endofunctors on  $\mathcal{G}$  to model the modalities of  $\text{LLMS}_c$ .

**Definition 3** ( $\dagger$  and  $G$  in  $\mathcal{G}$ ). *Consider the following endofunctors on  $\mathcal{G}$ :*

1. The functor  $\dagger$  on  $\mathcal{G}$  given on objects by

$$\dagger(U \leftarrow X) = (U \leftarrow X^*) \text{ where the relation } \dagger(\alpha) \text{ is defined by}$$

$$u(\dagger\alpha)(x_1, \dots, x_k) \text{ iff } u \alpha x_1 \text{ and } \dots \text{ and } u \alpha x_k$$

and  $\dagger$  acts on maps as  $\dagger(f, F) = (f, F^*)$ .

<sup>1</sup> There is another natural morphism  $\sigma_{A,B} : (A \odot B) \rightarrow (A \square B)$  where  $\square$  is the functor corresponding to the par connective, but it shall not concern us here.

2. The functor  $G$  on  $\mathcal{G}$  given by

$$G(U \leftarrow^\alpha X) = (U \leftarrow^{G\alpha} X^U)$$

where the relation  $G\alpha$  is given by  $u(G\alpha)f$  iff  $u\alpha f(u)$ . The functor  $G$  applied to a map  $(f, F): A \rightarrow B$  is  $(f, F(-)f): GA \rightarrow GB$ .

Formally the relation  $\dagger\alpha$  is given by the map  $U \times X^* \xrightarrow{\dagger\alpha} \mathbf{2}$  which is the composition

$$U \times X^* \xrightarrow{r_{U,X}} (U \times X)^* \xrightarrow{\alpha^*} \mathbf{2}^* \rightarrow \mathbf{2}$$

(the map  $r_{U,X}: U \times X^* \rightarrow (U \times X)^*$  takes a tuple  $\langle u, x_1, \dots, x_k \rangle \mapsto \langle \langle u, x_1 \rangle, \dots, \langle u, x_k \rangle \rangle$  and  $\mathbf{2}^* \rightarrow \mathbf{2}$  using the tensor product in  $\mathbf{2}$ ) and the relation  $G\alpha$  is given by the composition

$$U \times X^U \xrightarrow{\langle \pi_1, eval \rangle} U \times X \xrightarrow{\alpha} \mathbf{2}$$

**Lemma 3 (comonad structures for  $\dagger$  and  $G$ ).** *The functors  $\dagger$  and  $G$  have natural (monoidal) comonad structures  $(\dagger, \varepsilon_\dagger, \delta_\dagger)$  and  $(G, \varepsilon_G, \delta_G)$ , induced by the monad structure of  $*$  and the monads  $T_U$ , respectively.*

- The counit  $(\varepsilon_\dagger)_A: \dagger A \rightarrow A$  is given by identity on  $U$  and the singleton map  $X \xrightarrow{(\eta_*)X} X^*$  and the comultiplication  $(\delta_\dagger)_A: \dagger A \rightarrow \dagger^2 A$  is given by identity on  $U$  and forgetting brackets  $X^{**} \xrightarrow{(\mu_*)X} X^*$  in the second coordinate, as in the diagram

$$\begin{array}{ccc} U & \xleftarrow{\dagger\alpha} & X^* \\ id_U \downarrow & & \uparrow (\eta_*)X \\ U & \xleftarrow{\alpha} & X \end{array} \quad \begin{array}{ccc} U & \xleftarrow{\dagger\alpha} & X^* \\ id_U \downarrow & & \uparrow (\mu_*)X \\ U & \xleftarrow{\dagger^2\alpha} & X^{**} \end{array}$$

- The counit  $(\varepsilon_G)_A: GA \rightarrow A$  is given by identity on  $U$  and the natural constant map  $(\eta_{T_U})_X: X \rightarrow X^U$  in the second coordinate; the comultiplication  $(\delta_G)_A: GA \rightarrow G^2 A$  is given by identity on  $U$  and the restriction to the diagonal,  $(\mu_{T_U})_X: X^{U \times U} \rightarrow X^U$ , in the second coordinate:

$$\begin{array}{ccc} U & \xleftarrow{G\alpha} & X^U \\ id_U \downarrow & & \uparrow (\eta_{T_U})_X \\ U & \xleftarrow{\alpha} & X \end{array} \quad \begin{array}{ccc} U & \xleftarrow{G\alpha} & X^U \\ id_U \downarrow & & \uparrow (\mu_{T_U})_X \\ U & \xleftarrow{G^2\alpha} & X^{U \times U} \end{array}$$



Now there is a distributive law relating our comonads  $\dagger$  and  $G$ . Then, by abstract nonsense [2] we have that these comonads compose and that  $G$  lifts to the category of coalgebras for  $\dagger$ .

**Lemma 4 (distributivity law).** *There is a distributivity law of comonads  $\Lambda_A: G(\dagger A) \rightarrow \dagger(GA)$  given by*

$$\begin{array}{ccc} U & \xleftarrow{G(\dagger\alpha)} & (X^*)^U \\ \text{id}_U \downarrow & & \uparrow \lambda_X \\ U & \xleftarrow{\dagger(G\alpha)} & (X^U)^* \end{array}$$

where each component  $\lambda_X: (X^U)^* \rightarrow (X^*)^U$  of the natural transformation  $\lambda$  is given by the exponential transpose of the following composition

$$(X^U)^* \times U \xrightarrow{r_{X^U,U}} (X^U \times U)^* \xrightarrow{ev^*} X^*$$

**Definition 4 (definition of !).** *Consider the composite comonad  $(G; \dagger)$  and call it  $!$ . Its functor part acts on objects as  $!(U \xleftarrow{\alpha} X) = (U \xleftarrow{! \alpha} (X^*)^U)$  and on maps as  $!(f, F) = (f, F^*(-)f)$ . We have  $(\varepsilon!)_A: !A \rightarrow A$  and  $(\delta!)_A: !A \rightarrow !!A$  as follows:*

$$\begin{array}{ccc} U & \xleftarrow{! \alpha} & (X^*)^U \\ \text{id}_U \downarrow & & \uparrow \eta_X \\ U & \xleftarrow{\alpha} & X \end{array} \quad \begin{array}{ccc} U & \xleftarrow{! \alpha} & (X^*)^U \\ \text{id}_U \downarrow & & \uparrow \mu_X \\ U & \xleftarrow{!^2 \alpha} & (((X^*)^U)^*)^U \end{array}$$

where  $\eta_X$  is given by  $X \xrightarrow{(\eta_*)^X} X^* \xrightarrow{(\eta_U)^X} (X^*)^U$  and  $\mu_X$  is given by the exponential transpose of the composition  $((X^*)^U)^* \times U \xrightarrow{(ev, \pi)} ((X^*)^U)^* \times U \xrightarrow{\lambda \times U} (X^{**})^U \times U \xrightarrow{ev} X^{**} \xrightarrow{(\mu_*)^X} X^*$ .

As the name indicates it is this composite comonad that will play the role of the modality  $!$  in Linear Logic. Next we emphasize some properties which are used in the proof of soundness of the model.

**Proposition 2 (interplay of ! and  $\dagger$ ).** *We have the following facts*

1. There is a comonad morphism  $^2 \kappa: ! \rightarrow \dagger$ , such that  $\kappa_A: !A \rightarrow \dagger A$  is given by

$$\begin{array}{ccc} U & \xleftarrow{! \alpha} & (X^*)^U \\ \text{id}_U \downarrow & & \uparrow \beta_X \\ U & \xleftarrow{\dagger \alpha} & X^* \end{array}$$

where  $\beta_X: X^* \rightarrow (X^*)^U$  is given by  $(\eta_{T_U})$  applied to the object  $X^*$ .

2. There is a natural morphism  $\dagger A \otimes \dagger B \rightarrow \dagger(A \otimes B)$  given by  $(\text{id}_{U \times V}, (\lambda_1, \lambda_2) \circ r)$  as the diagram shows

$$\begin{array}{ccc} U \times V & \xleftarrow{\dagger(\alpha) \otimes \dagger(\beta)} & (X^*)^V \times (Y^*)^U \\ \text{id}_{U \times V} \downarrow & & \uparrow (\lambda_1, \lambda_2) \circ r \\ U \times V & \xleftarrow{\dagger(\alpha \otimes \beta)} & (X^V \times Y^U)^* \end{array}$$

where, intuitively, the map  $r: (X^V \times Y^U)^* \rightarrow (X^V)^* \times (Y^U)^*$  transforms a sequence  $\langle f_1 \times g_1 \cdots f_k \times g_k \rangle$  into pairs of sequences  $(\langle f_1 \cdots f_k \rangle, \langle g_1 \cdots g_k \rangle)$ ; and the maps  $\lambda_1: (X^V)^* \rightarrow (X^*)^V$  and  $\lambda_2: (Y^U)^* \rightarrow (Y^*)^U$  are components of natural transformations which satisfies the conditions for a “distributive law of monads”

3. For any  $\mathcal{G}$ -object  $A$ ,  $!A$  has a comonoid structure  $(!A, d_A, e_A)$  with respect to  $\otimes$  with  $e_A: !A \rightarrow I$  and  $d_A: !A \rightarrow !A \otimes !A$  as its counit and comultiplication, respectively. There is a functor from  $\mathcal{G}$  to  $\mathbf{Comon}_{\otimes} \mathcal{G}$  which takes  $A$  to  $(!A, d_A, e_A)$ .
4. For any  $\mathcal{G}$ -object  $A$ ,  $\dagger A$  has a comonoid structure  $(\dagger A, d'_A, e'_A)$  with respect to  $\odot$  with  $e'_A: \dagger A \rightarrow I$  and  $d'_A: \dagger A \rightarrow \dagger A \odot \dagger A$  as its counit and comultiplication, respectively. There is a functor  $R_{\dagger}: \mathcal{G} \rightarrow \mathbf{Comon}_{\odot} \mathcal{G}$  which takes  $A$  to  $(\dagger A, d'_A, e'_A)$ . Moreover  $R_{\dagger}$  is right-adjoint to the forgetful functor  $U: \mathbf{Comon}_{\odot} \mathcal{G} \rightarrow \mathcal{G}$ .

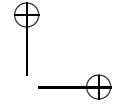
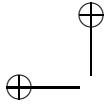
Having assembled all the necessary pieces we can now put them together in the:

**Theorem 2 (Soundness of  $\mathcal{G}$ ).** *The symmetric monoidal closed category  $\mathcal{G}$ , with bifunctors internal-hom  $[-, -]_{\mathcal{G}}$ ; tensor products  $\otimes$  and  $\odot$ ; the additive conjunction  $\&$  and the comonads  $!$  and  $\dagger$ , is a model for  $\text{LLMS}_c$ . Thus, each entailment  $\Gamma \vdash_{\text{LLMS}_c} A$  corresponds to the existence of a morphism in  $\mathcal{G}$ ,  $(f, F): \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket$ .*

To prove the theorem, we have only to check that the rules of  $\text{LLMS}_c$  are preserved by the interpretation presented.

As we hinted before, the category has more structure than the necessary to model  $\text{LLMS}_c$ : for example we do have multiplicative and additive disjunctions in  $\mathcal{G}$ , as well as a linear negation.

<sup>2</sup> This morphism is similar to Reddy’s Ser, but it is not a monomorphism in  $\mathcal{G}$ .



#### 4 Back to LLMS

Two years after the work on  $LLMS_c$ , I realized that the modelling using dialectica categories could be extended for non-commutative systems like LLMS. My doctoral student then, Alexander Tucker, wrote a short note on his investigations of the non-commutative dialectica model and published it in the ESSLLI Student Session proceedings [14]. But Alex decide to stop working on his doctorate and hence this work was left untouched for several years. We recall some of Tucker’s results.

Reddy first set out the criteria he believed necessary for a categorical model of LLMS. He starts with a symmetric monoidal closed category  $(\mathcal{C}, \otimes, I, -\circ)$  together with an additional monoidal structure  $(\triangleright, I)$  such that  $(\otimes, I)$  is a sub-monoidal structure of  $(\triangleright, I)$ . The last condition means that there is a natural monic  $\text{ser}: A \otimes B \rightarrow A \triangleright B$  which preserves the associated monoidal structure. He then adds two comonads associated to the tensorial structures given by  $\otimes$  and  $\triangleright$ .

Given such a set up, Reddy defined a LLMS-*category* as a symmetric monoidal closed category, with products and co-products, together with the extra tensor product  $\triangleright$  and two monoidal comonads,  $!$  and  $\dagger$  with the former being a sub-comonad of the latter via a comonad monomorphism  $\text{Ser}: ! \rightarrow \dagger$ . So the intuitio is that  $\dagger$  transforms  $\triangleright$  tensors into  $\otimes$  tensors and  $!$  (as usual) transforms tensors into cartesian products. But also  $!A$  must be a comonoid with respect to  $\otimes$  and  $\dagger A$  a comonoid with respect to  $\triangleright$ . Then since  $\otimes$  is a sub-monoidal structure of  $\triangleright$ ,  $!A$  becomes a comonoid with respect to  $\triangleright$  via the monic  $\text{ser}$ . This comonoid must be a sub-comonoid of  $\dagger A$  via the natural monic  $\text{Ser}$  which must now be a morphism of comonoids as well.

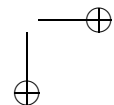
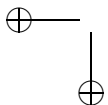
This definition, generalized from the example of coherence spaces, needs to be taken with a pinch of salt. Further work by Bechet, de Groote and Retoré [1] showed that the two kinds of logical context, represented by the operations of  $\otimes$  and  $\triangleright$  need a more sophisticated treatment.

The partial order over the contexts of the LLMS calculus are in fact a class of order called *series parallel orders* or SP-orders. These are defined as the least class of partial orders containing the single element order, and closed under disjoint union and ordinal sum.

In Bechet et al [1] a set of rewrite rules is given which is a complete characterization of the inclusion of one SP-order in another and so by extension corresponds exactly to Reddy’s  $\text{Ser}$  rule. In fact, the non-trivial rewrite rules correspond precisely to the four rules given above. Moreover, if we insist that both  $\otimes$  and  $\triangleright$  have identities which are equal, that  $\otimes$  is associative and commutative and that  $\triangleright$  is associative and not commutative, then the set of rewrite rules are all derivable from just the middle interchange rule.

The natural transformation  $\text{ser}$  can be obtained from  $\text{int}$  by instantiating  $B$  and  $C$  as  $I$ ,  $\text{wd1}$  can be obtained by instantiating  $B$  as  $I$  and  $\text{wd2}$  by instantiating  $C$  as  $I$ . We therefore need to exchange  $\text{ser}$  for  $\text{int}$  in the definition of an LLMS-category. With this big difference in setting, we can re-state Reddy’s definition of a LLMS-category.

We relax Reddy’s requirements somewhat by neither insisting on monotonicity of the natural transformations, nor dealing with the additive constructs. We also exchange the natural transformation  $\text{ser}$  with a natural transformation  $\text{int}$  corresponding to the middle interchange rule.



**Definition 5 (Tucker).** A modality-free LLMS-category is an smcc  $(\mathcal{C}, \otimes, -\circ, I)$  together with an additional monoidal structure  $(\triangleright, I)$ . The two structures are related via a monoidal natural transformation:  $\text{int} : (A \triangleright B) \otimes (C \triangleright D) \rightarrow (A \triangleright C) \otimes (B \triangleright D)$ .

An LLMS-category is then a modality-free LLMS-category along with two monoidal comonads  $!$  and  $\dagger$  related by a comonad morphism  $\text{Ser} : ! \rightarrow \dagger$ . Also  $!A$  must be a comonoid with respect to  $(\otimes, I)$ ,  $\dagger A$  a comonoid with respect to  $(\triangleright, I)$  and  $\text{Ser}$  a morphism of comonoids.

With this new definition in tow we can provide a simple dialectica model for the system LLMS itself, instead of its commutative version LLMS<sub>c</sub>. As before, our categorical model of LLMS is based on a simple dialectica category over the category of sets, but this time the objects are partial relations over a *three* valued poset, or maps into  $\{\text{false}, \text{undefined}, \text{true}\}$ .

**Definition 6 (dialectica category  $\mathcal{L}$ ).** The category  $\mathcal{L}$  is described by:

- Objects are partial relations on **Sets**, that is maps  $U \times X \xrightarrow{\alpha} \mathbf{3}$ , written as  $(U \xleftarrow{\alpha} X)$ .
- A morphism from  $(U \xleftarrow{\alpha} X)$  to  $(V \xleftarrow{\beta} Y)$  consists a pair of maps,  $f : U \rightarrow V$  and  $F : Y \rightarrow X$ , such that the following condition is satisfied

$$\alpha(u, Fy) \leq \beta(fu, y)$$

where  $\leq$  is the usual order in  $\mathbf{3}$ .

With easy, suitable modifications of the work in the previous section, we can describe the structure of  $\mathcal{L}$ .

**Definition 7 (structure of  $\mathcal{L}$ ).** Let  $A = (U \xleftarrow{\alpha} X)$  and  $B = (V \xleftarrow{\beta} Y)$  be objects of the category  $\mathcal{L}$ .

- The tensor bifunctor  $\otimes : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}$  given by

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

where  $(u, v) \alpha \otimes \beta (f, g)$  iff  $u \alpha f(v)$  and  $v \beta g(u)$  still has as its unit the object  $I = (\mathbf{1} \xleftarrow{\iota} \mathbf{1})$ , where  $\iota$  is the identity relation on  $\mathbf{1}$ .

- The internal-hom bifunctor  $[-, -]_{\mathcal{G}} : \mathcal{G}^{op} \times \mathcal{G} \rightarrow \mathcal{G}$  is still given by

$$[A, B]_{\mathcal{G}} = (V^U \times X^Y \xleftarrow{\beta^\alpha} U \times Y),$$

where  $(f, F) \beta^\alpha (u, y)$  iff  $u \alpha x$  implies  $v \beta y$ . The relation  $\beta^\alpha$  is given by the composition

$$V^U \times X^Y \times U \times Y \xrightarrow{(\pi_3, \pi_4, \text{eval}, \text{eval})} U \times Y \times V \times X \xrightarrow{\alpha \times \beta'} \mathbf{3} \times \mathbf{3} \xrightarrow{-\circ} \mathbf{3}.$$

- There is a non-commutative tensor bifunctor  $\triangleright : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}$  given by

$$A \triangleright B = (U \times V \xleftarrow{\alpha \triangleright \beta} X \times Y),$$

where  $(u, v) \alpha \triangleright \beta (x, y)$  iff  $u \alpha x$  land  $v \beta y$ , where land is the computer science lazy and. This connective is non-commutative, since **false** land **undefined** is **false**, but **undefined** land **false** is **undefined**. The unit of this tensor product is the same object  $I$  that we had before.

Using the same definitions we had before for comonads  $!$  and  $\dagger$ , except that now, instead of *commutative* free monoids, we use simply free monoids, we can model the system LLMS modalities and obtain.

**Theorem 3 (Soundness of  $\mathcal{L}$ ).** *The symmetric monoidal closed category  $\mathcal{L}$ , with bifunctors internal-hom  $[-, -]_{\mathcal{L}}$ ; tensor products  $\otimes$  and  $\triangleright$ ; and the comonads  $!$  and  $\dagger$ , is a model for LLMS. Thus, each entailment  $\Gamma \vdash_{\text{LLMS}} A$  corresponds to the existence of a morphism in  $\mathcal{L}$ ,  $(f, F): \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket$ .*

## 5 Conclusions

The motivation for this work came from O’Hearn’s and Reddy’s use of Linear Logic to deal with sequentiality in the semantics of Algol-like programming languages. Earlier on we realised that we could use Reddy’s ideas to give a syntactical characterization to some of the (somewhat mysterious) constructions of dialectica categorical models  $\text{Dia}_2(\text{Set})$  for Classical Linear Logic. In particular it was interesting to find an intuitive explanation for the comonad  $\dagger$  in terms of the extra tensor product  $\odot$ .

Summing up the work so far, we adapted Reddy’s system LLMS providing it with a commutative version of the connective “before” and an associated modality. A dialectica category  $\mathcal{G}$  was constructed on **Sets** and shown to be a sound model of the new system  $\text{LLMS}_c$ . Then we went back to the drawing board, in the light of work of Bechet, de Groote and Retoré and uncovered a dialectica model of LLMS itself.

Much work remains to be done. Firstly, going back to the original motivation, more work is needed to relate the system  $\text{LLMS}_c$  to O’Hearn’s and Reddy’s other work on Algol-like languages [10, 12]. Secondly, from a logical perspective, we might want to use this system with two modalities, as a blueprint for work on relating modalities in general. Thirdly most of the recent work on sequentiality [3] uses games and *polarized* linear logic. It would be nice if we could find a mathematically significant relationship between linear logic polarization and dialectica categories.

## References

1. D. Bechet, Ph. de Groote, and C. Retoré. A complete axiomatisation for the inclusion of series-parallel partial orders. In *International Conference on Rewriting Techniques and Applications*, Sitges, Spain, October 1997.
2. J. Beck. Distributive laws. In *Seminar on Triples and Categorical Homoly Theory, LNM 80*, pages 119–140. Springer-Verlag, 1969.
3. M. Churchill and J. Laird. A logic of sequentiality. In *Computer Science Logic (Lecture Notes in Computer Science)*. Springer Verlag, pp. 215-229., Brno, Czech Republic, August 2010.
4. M. da S. Corrêa, E. H. Haeusler, and V. C. V. de Paiva. A dialectica model of state. Technical Report PUC-Rio.MCC21/95, Dept. de Informática, Pontifícia Univ. Católica PUC-Rio, August 1995. (Electronically available in [ftp://ftp.inf.puc-rio.br/pub/docs/techreports/95\\_21\\_correa.ps.gz](ftp://ftp.inf.puc-rio.br/pub/docs/techreports/95_21_correa.ps.gz)).
5. M. da S. Corrêa, E. H. Haeusler, and V. C. V. de Paiva. A dialectica model of state. In *CATS’96, Computing: The Australian Theory Symposium Proceedings*, Melbourne, Australia, January 1996.

6. V. C. V de Paiva. Dialectica categories. In *Categories in Computer Science and Logic*, Boulder, Colorado, June 1987. American Mathematical Society. (Contemporary Mathematics, Vol. 92).
7. V. C. V de Paiva. *Dialectica Categories*. PhD thesis, Lucy Cavendish College, Cambridge, November 1988. Printed as Technical Report No. 213, University of Cambridge Computer Laboratory, January, 1991.
8. V. C. V. de Paiva. A dialectica-like model of linear logic. In *Category Theory in Computer Science*, volume 389 of *LNCS*, pages 341–356, September 1989.
9. P. W. O’Hearn. Linear logic and interference control. In *Categories in Computer Science and Logic*, volume 530 of *LNCS*, pages 74–93, Paris, France, September 1991. Springer-Verlag, Berlin.
10. P. W. O’Hearn, A. J. Power, M. Takeyama, and R. D. Tennent. Syntactic control of interference revisited. In *MFPS XI*, volume 1 of *Electronic Notes in Theoretical Computer Science*, New Orleans, March 1995.
11. U. S. Reddy. A linear logic model of state: Preliminary report. Electronic manuscript (ftp).csc.uiuc.edu, directory /pub/reddy/papers, May 1993.
12. U. S. Reddy. Passivity and independence. In *9th Annual IEEE Symposium on Logic in Computer Science*, pages 342–352, Paris, France, 1994. IEEE Computer Society Press, Los Alamitos, California.
13. C. Retoré. *Reseaux et Séquents Ordonnés Mathématiques*. PhD thesis, Université Paris 7, Février 1993.
14. A. Tucker. Categorical modelling of a state oriented version of linear logic. In *Proceedings of the Third ESSLLI Student Session*, Saarbruecken, Germany, August 1998.