Linear logic model of state revisited

VALERIA DE PAIVA*, *Nuance Communications, Sunnyvale, CA 94085, USA*

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In an unpublished note Reddy introduced an extended intuitionistic linear calculus, called LLMS (for Linear Logic Model of State), to model state manipulation via the notions of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. Earlier and independently de Paiva introduced a (collection of) dialectica categorical models for (classical and intuitionistic) Linear Logic, the categories $\mathsf{Dial}_2\mathsf{Set}$. These categories contain, apart from the structure needed to model linear logic, an extra tensor product functor and an extra comonad structure corresponding to a modality related to the extra tensor product. It is surprising that these works arising from completely different motivations can be related in a meaningful way. In this article, following joint work with Corrêa and Haeusler, we first adapt Reddy's system LLMS providing a commutative version of the connective 'before' and its associated modality and then construct a dialectica category on **Sets**, which we show is a sound model for the modified version of Reddy's the system LLMS_c. Moreover, following the work of Tucker, we provide another variant of the Dialectica categories with a non-commutative tensor and its 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.

1 Motivation

This paper is 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, which makes it hard to find and hard to read. Similarly, Tucker's work described [19] was only published on proceedings of the doctoral students workshop of ESSLLI 1998. But work on modelling state manipulation using linear logic insights is still flourishing. Mostly via the connection to game semantics and the work of G. McCusker, J. Laird, P. O'Hearn and U. Reddy himself amongst others, many ideas are still being being debated and improved upon. Thus it seems reasonable to look again at what the connections (via categorical models) of these systems can tell us about sequentiality and the incorporation of state manipulation in functional programming languages.

2 Introduction

The concept of state-manipulation plays an important role in computation. There are many attempts to formalize this concept especially using type systems derived from Girard's Linear Logic ([12, 14]). 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 [14]. 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

*E-mail: valeria.depaiva@gmail.com

Vol. 22 No. 5, © The Author 2014. Published by Oxford University Press. All rights reserved.
For Permissions, please email: journals.permissions@oup.com
doi:10.1093/jigpal/jzu013 Advance Access published 22 May 2014

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 embbeding 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 non-commutative. 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 communication in both ways here, but it can be said to represent concurrent execution without interaction or synchronization. In this article, 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 $\mathsf{Dial}_2(Set)$. The categories $\mathsf{Dial}_2(Set)$ have exactly the extra structure (an extra tensor product and an extra modality, associated with this tensor product) to model 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 3) describe the proof system LLMS_c, obtained by modifying LLMS to make the connective 'before' commutative. In Section 4, we describe the specific dialectica $\mathcal{G}$ category (built over Sets) and show that it is a sound model for LLMS_c. Then we recap the work of Tucker [19] where a variation of the construction produces a model of LLMS itself. Finally, we conclude pointing out some future work.

3 The system LLMS_c

The presentation of the system of LLMS_c follows strictly the presentation of LLMS [14], which itself follows from Retoré's work [16] on *pomset logic*.

The left context of a sequent in LLMS_c has the following syntax:

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

where ϵ (empty context) is interpreted as $\mathbf{1}$ and the contexts Γ_0, Γ_1 and $\Gamma_0; \Gamma_1$ are interpreted as $\Gamma_0 \otimes \Gamma_1$ and $\Gamma_0 \odot \Gamma_1$, respectively. A context Γ, which does not contain any occurrences of the connective ';', is called an *independent context*.

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

$$\begin{array}{rl}
|\epsilon| = \{\} & \\
|A| = \{A\} & \leq_A = \{(A,A)\} \\
|\Gamma_1,\Gamma_2| = |\Gamma_1| + |\Gamma_2| & \leq_{\Gamma_1,\Gamma_2} = \leq_{\Gamma_1} \cup \leq_{\Gamma_2} \\
|\Gamma_1;\Gamma_2| = |\Gamma_1| + |\Gamma_2| & \leq_{\Gamma_1;\Gamma_2} = \leq_{\Gamma_1} \cup \leq_{\Gamma_2} \cup (|\Gamma_1| \times |\Gamma_2|) \cup (|\Gamma_2| \times |\Gamma_1|)
\end{array}$$

The rules of LLMS_c, are just those of LLMS [14], 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 LLMS_c are presented in the Figure 1.

Structural Rule

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

Identity Rules

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

Multiplicative Rules

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

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

Additive Rules

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

Modalities

$$\frac{\Gamma[\epsilon] \vdash C}{\Gamma[\dagger A] \vdash C}\,\dagger\mathbf{Weak} \qquad \frac{\Gamma[A] \vdash C}{\Gamma[\dagger A] \vdash C}\,\dagger\mathbf{Der} \qquad \frac{\Gamma[\dagger A; \dagger A] \vdash C}{\Gamma[\dagger A] \vdash C}\,\dagger\mathbf{Thread}$$

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

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

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

FIG. 1. Proof system LLMS_c.

Figure 1 also works as a representation of the rules of LLMS, if we replace in the definition of contexts the last rules by:

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

3.1 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 3.1
The following inferences are allowed by the Structural Rule (**Ser**).

$$\textbf{(i)}\frac{\Delta[(A,B);C]\vdash D}{\Delta[A,(B;C)]\vdash D} \quad \textbf{(ii)}\frac{\Delta[A;(B,C)]\vdash D}{\Delta[(A;B),C]\vdash D}$$

$$\textbf{(iii)}\frac{\Delta[A;B]\vdash C}{\Delta[A,B]\vdash C} \quad \textbf{(iv)}\frac{\Delta[(A,C);(B,D)]\vdash E}{\Delta[(A;B),(C;D)]\vdash E}$$

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

$$A\otimes(B\odot C) \multimap (A\otimes B)\odot C$$
$$(A\odot B)\otimes C \multimap A\odot(B\otimes C)$$

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

$$A\otimes B \multimap A\odot B$$
$$(A\odot B)\otimes(C\odot D) \multimap (A\otimes C)\odot(B\otimes D)$$

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

ser : $A\otimes B \rightarrow A\odot B$

wd1 : $A\otimes(B\odot C) \rightarrow (A\otimes B)\odot C$

wd2 : $(A\odot B)\otimes C \rightarrow A\odot(B\otimes C)$

int : $(A\odot B)\otimes(C\odot D) \rightarrow (A\odot C)\otimes(B\odot D)$

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.

3.2 Cut elimination for LLMS

We can obtain a Gentzen's style proof of the cut-elimination theorem for LLMS_c.

THEOREM 3.2 (Cut Elimination)
If a sequent is provable in LLMS_c, then it is provable in LLMS_c without an application of the cut rule.

This proof is probably similar to Reddy's proof [14], 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 payed to the cases involving †, for instance, the case in which the †**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 of cut elimination can be found in the technical report [4].

4 A categorical model for LLMS_c

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

Recall that the partial-order with two elements **2**, where one thinks of **0** as false and **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 linear internal-hom is the usual Heyting implication.

DEFINITION 4.1 (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 $(U \overset{\alpha}{\leftarrow\!\!\!+\!\!\!-} X)$.
- A morphism from $(U \overset{\alpha}{\leftarrow\!\!\!+\!\!\!-} X)$ to $(V \overset{\beta}{\leftarrow\!\!\!+\!\!\!-} Y)$ consists a pair of maps, $f\colon U \to V$ and $F\colon Y \to X$, such that the following condition is satisfied

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

  where $\leq$ is the usual order in **2**.

Thus there is a morphism (f, F) from α to β 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:

$$\begin{tikzcd}
U \arrow[d, "f"'] & X \arrow[l, "\alpha"'] \\
V & Y \arrow[l, "\beta"] \arrow[u, "F"']
\end{tikzcd}$$

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

DEFINITION 4.2 (structure of $\mathcal{G}$)
Let $A = (U \overset{\alpha}{\leftarrow\!\!\!+\!\!\!-} X)$ and $B = (V \overset{\beta}{\leftarrow\!\!\!+\!\!\!-} Y)$ be objects of the category $\mathcal{G}$.

- The tensor bifunctor $\otimes\colon \mathcal{G}\times\mathcal{G}\to\mathcal{G}$ is given by

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

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

- The tensor bifunctor $\odot\colon \mathcal{G}\times\mathcal{G}\to\mathcal{G}$ is given by

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

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 $[-,-]_{\mathcal{G}}\colon \mathcal{G}^{op}\times\mathcal{G}\to\mathcal{G}$ is given by

$$[A,B]_{\mathcal{G}}=(V^U\times X^Y \overset{\beta^\alpha}{\leftarrow\!\!\!+\!\!\!-} U\times Y),$$

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

$$V^U\times X^Y\times U\times Y \xrightarrow{\langle\pi_3,\pi_4,eval,eval\rangle} U\times Y\times V\times X \xrightarrow{`\alpha\times\beta'} \mathbf{2}\times\mathbf{2}\xrightarrow{-\!\circ}\mathbf{2}.$$

- The product bifunctor $\&\colon \mathcal{G}\times\mathcal{G}\to\mathcal{G}$ is given by

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

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 $\top=(\mathbf{1}\overset{e}{\leftarrow\!\!\!+\!\!\!-}\mathbf{0})$.

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 4.3 ($\mathcal{G}$ 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 4.4 ($\mathcal{G}$ natural transformations)
For any A and B objects of $\mathcal{G}$ there are natural morphisms:

(i) $\mathsf{ser}_{(A,B)}\colon A\otimes B\to A\odot B$; [1]
(ii) $\mathsf{wd1}_{(A,B,C)}\colon A\otimes(B\odot C)\to(A\otimes B)\odot C$ (called weak-distributivity 1)
(iii) $\mathsf{wd2}_{(A,B,C)}\colon (A\odot B)\otimes C\to A\odot(B\otimes C)$ (called weak-distributivity 2)
(iv) $\mathsf{int}_{(A,B,C,D)}\colon (A\odot B)\otimes(C\odot D)\to(A\otimes C)\odot(B\otimes D)$. (called middle interchange law)

Following the pattern of linear logic, modelling the modalities of LLMS_c, is more complicated than describing its tensor products. Recall that the category **Sets** does have free (commutative) monoid structures. There is an adjunction

$$F\dashv \mathrm{U}\colon \mathbf{Mon}_c\to\mathbf{Sets}$$

[1]There is another natural morphism $\sigma_{A,B}\colon (A\odot B)\to(A\Box B)$ where $\Box$ is the functor corresponding to the par connective, but it shall not concern us here.

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, $e_X: \mathbf{1} \to X^*$ is the empty sequence and $m_X: X^* \times X^* \to X^*$ 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: **Sets** $\to$ **Sets** is the endofunctor which takes $X \mapsto X^U$, $Y \mapsto Y^U$ and if $g: X \to Y, f \in X^U$ is taken to $f;g \in Y^U$. Its unit is given by the transpose of the projection, that is the constant map $(\eta_{T_U})_X: X \to X^U$ and its multiplication $(\mu_{T_U})_X: X^{U\times U} \to X^U$, is simply precomposition with the diagonal map on U.

We define two endofuctors on $\mathcal{G}$ to model the modalities of LLMS_c.

Definition 4.5 († and G in $\mathcal{G}$)
Consider the following endofunctors on $\mathcal{G}$:

1. The functor † on $\mathcal{G}$ given on objects by
   $\dagger(U \overset{\alpha}{\leftarrow\!\!\!+\!\!\!\rightarrow} X) = (U \overset{\dagger\alpha}{\leftarrow\!\!\!+\!\!\!\rightarrow} X^*)$ where the relation $\dagger(\alpha)$ is defined by

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

   and † acts on maps as $\dagger(f,F)=(f,F^*)$.
2. The functor G on $\mathcal{G}$ given by

$$G(U \overset{\alpha}{\leftarrow\!\!\!+\!\!\!\rightarrow} X) = (U \overset{G\alpha}{\leftarrow\!\!\!+\!\!\!\rightarrow} X^U)$$

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

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

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

(the map $r_{U,X}: U \times X^* \to (U \times X)^*$ takes a tuple $\langle u, x_1, \ldots, x_k\rangle \mapsto \langle\langle u, x_1\rangle, \ldots, \langle u, x_k\rangle\rangle$ and $\mathbf{2}^* \to \mathbf{2}$ 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 4.6 (comonad structures for † and G)
The functors † and G have natural (monoidal) comonad structures $(\dagger, \epsilon_\dagger, \delta_\dagger)$ and $(G, \epsilon_G, \delta_G)$, induced by the monad structure of $*$ and the monads T_U, respectively.

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

$$\begin{tikzcd}
U \arrow[d, "id_U"'] & X^* \arrow[l, "\dagger\alpha"', "\shortmid"{marking}] \\
U & X \arrow[l, "\alpha"', "\shortmid"{marking}] \arrow[u, "(\eta_*)_X"']
\end{tikzcd}
\qquad
\begin{tikzcd}
U \arrow[d, "id_U"'] & X^* \arrow[l, "\dagger\alpha"', "\shortmid"{marking}] \\
U & X^{**} \arrow[l, "\dagger^2\alpha"', "\shortmid"{marking}] \arrow[u, "(\mu_*)_X"']
\end{tikzcd}$$

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

$$\begin{tikzcd}
U \arrow[d, "id_U"'] & X^U \arrow[l, "G\alpha"', "\shortmid"{marking}] \\
U & X \arrow[l, "\alpha"', "\shortmid"{marking}] \arrow[u, "(\eta_{T_U})_X"']
\end{tikzcd}
\qquad
\begin{tikzcd}
U \arrow[d, "id_U"'] & X^U \arrow[l, "G\alpha"', "\shortmid"{marking}] \\
U & X^{U\times U} \arrow[l, "G^2\alpha"', "\shortmid"{marking}] \arrow[u, "(\mu_{T_U})_X"']
\end{tikzcd}$$

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

Lemma 4.7 (distributivity law)
There is a *distributivity law* of comonads $\Lambda_A\colon G(\dagger A) \to \dagger(GA)$ given by

$$\begin{tikzcd}
U \arrow[d, "id_U"'] & (X^*)^U \arrow[l, "G(\dagger\alpha)"', "\shortmid"{marking}] \\
U & (X^U)^* \arrow[l, "\dagger(G\alpha)"', "\shortmid"{marking}] \arrow[u, "\lambda_X"']
\end{tikzcd}$$

where each component $\lambda_X\colon (X^U)^* \to (X^*)^U$ of the natural transformation λ 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 4.8 (definition of !)
Consider the composite comonad $(G;\dagger)$ and call it !. Its functor part acts on objects as $!(U \overset{\alpha}{\leftarrow} X) = (U \overset{!\alpha}{\leftarrow} (X^*)^U)$ and on maps as $!(f,F) = (f, F^*(-)f)$. We have $(\epsilon_!)_A\colon\, !A \to A$ and $(\delta_!)_A\colon\, !A \to !!A$ as follows:

$$\begin{tikzcd}
U \arrow[d, "id_U"'] & (X^*)^U \arrow[l, "!\alpha"', "\shortmid"{marking}] \\
U & X \arrow[l, "\alpha"', "\shortmid"{marking}] \arrow[u, "\eta_X"']
\end{tikzcd}
\qquad
\begin{tikzcd}
U \arrow[d, "id_U"'] & (X^*)^U \arrow[l, "!\alpha"', "\shortmid"{marking}] \\
U & (((X^*)^U)^*)^U \arrow[l, "!^2\alpha"', "\shortmid"{marking}] \arrow[u, "\mu_X"']
\end{tikzcd}$$

where η_X is given by $X \overset{(\eta_*)_X}{\to} X^* \overset{(\eta_{T_U})_X}{\to} (X^*)^U$ and μ_X is given by the exponential transpose of the composition $(((X^*)^U)^*)^U \times U \overset{\langle ev,\pi\rangle}{\longrightarrow} ((X^*)^U)^* \times U \overset{\lambda\times U}{\longrightarrow} (X^{**})^U \times U \overset{ev}{\to} X^{**} \overset{(\mu_*)_X}{\to} X^*$.

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 4.9 (interplay of ! and †)
We have the following facts

1. There is a comonad morphism [2] $\kappa\colon ! \to \dagger$, such that $\kappa_A\colon !A \to \dagger A$ is given by

$$\begin{tikzcd} U & (X^*)^U \arrow[l, "!\alpha"'] \\ U \arrow[from=u, "id_U"'] & X^* \arrow[l, "\dagger\alpha"] \arrow[u, "\beta_X"'] \end{tikzcd}$$

   where $\beta_X\colon X^* \to (X^*)^U$ is given by (η_{T_U}) applied to the object X^*.
2. There is a natural morphism $\dagger A \otimes \dagger B \to \dagger(A\otimes B)$ given by $(id_{U\times V}, (\lambda_1,\lambda_2)\circ r)$ as the diagram shows

$$\begin{tikzcd} U\times V & (X^*)^V \times (Y^*)^U \arrow[l, "\dagger(\alpha)\otimes\dagger(\beta)"'] \\ U\times V \arrow[from=u, "id_{U\times V}"'] & (X^V\times Y^U)^* \arrow[l, "\dagger(\alpha\otimes\beta)"] \arrow[u, "(\lambda_1,\lambda_2)\circ r"'] \end{tikzcd}$$

   where, intuitively, the map $r\colon (X^V\times Y^U)^* \to (X^V)^* \times (Y^U)^*$ 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 $\lambda_1\colon (X^V)^* \to (X^*)^V$ and $\lambda_2\colon (Y^U)^* \to (Y^*)^U$ 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 $e_A\colon !A \to I$ and $d_A\colon !A \to !A\otimes !A$ 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 $e'_A\colon \dagger A \to I$ and $d'_A\colon \dagger A \to \dagger A \odot \dagger A$ as its counit and comultiplication, respectively. There is a functor $R_\dagger\colon \mathcal{G} \to \mathbf{Comon}_\odot\mathcal{G}$ which takes A to $(\dagger A, d'_A, e'_A)$. Moreover $R_\dagger$ is right-adjoint to the forgetful functor U: $\mathbf{Comon}_\odot\mathcal{G} \to \mathcal{G}$.

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

Theorem 4.10 (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 †, is a model for LLMS_c. Thus, each entailment $\Gamma \vdash_{\mathsf{LLMS}_c} A$ corresponds to the existence of a morphism in $\mathcal{G}$, $(f,F)\colon [\![\Gamma]\!] \to [\![A]\!]$.

[2] This morphism is similar to Reddy's Ser, but it is not a monomorphism in $\mathcal{G}$.

To prove the theorem, we have only to check that the rules of LLMS_c are preserved by the interpretation presented.

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

5 Back to LLMS

Two years after the work on LLMS_c, I realized that the model using dialectica categories could be extended to non-commutative systems like LLMS. Alexander Tucker, at that time my doctoral student, wrote a short note on his investigations of the non-commutative dialectica model and published it in the ESSLLI Student Session proceedings [19]. But Alex decided to stop working on his doctorate and his work was left untouched for several years. We briefly recall 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 $(\mathcal{C}, \otimes, I, \multimap)$ together with an additional monoidal structure $(\rhd, I)$ such that $(\otimes, I)$ is a sub-monoidal structure of $(\rhd, I)$. The last condition means that there is a natural monic $\mathsf{ser}\colon A \otimes B \to A \rhd B$ which preserves the associated monoidal structure. He then adds two comonads associated to the tensorial structures given by $\otimes$ and $\rhd$.

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 $\rhd$ and two monoidal comonads, ! and † with the former being a sub-comonad of the latter via a comonad monomorphism $\mathsf{Ser}\colon ! \to \dagger$. So the intuition is that † transforms $\rhd$ 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 $\rhd$. Then since $\otimes$ is a sub-monoidal structure of $\rhd$, $!A$ becomes a comonoid with respect to $\rhd$ via the monic ser. This comonoid must be a sub-comonoid of $\dagger A$ via the natural monic 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 $\rhd$ 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 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 $\rhd$ have identities which are equal, that $\otimes$ is associative and commutative and that $\rhd$ is associative and not commutative, then each rule in the set of rewrite rules is derivable from just the middle interchange rule.

The natural transformation ser can be obtained from int by instantiating B and C as I, $\mathsf{wd1}$ can be obtained by instantiating B as I and $\mathsf{wd2}$ by instantiating C as I. We therefore need to exchange ser for 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 ser by a natural transformation int corresponding to the middle interchange rule.

DEFINITION 5.1 (Tucker)
A modality-free LLMS-category is an smcc $(\mathcal{C}, \otimes, \multimap, I)$ together with an additional monoidal structure $(\triangleright, I)$. The two structures are related via a monoidal natural transformation: $\mathsf{int}\colon (A \triangleright B) \otimes (C \triangleright D) \to (A \triangleright C) \otimes (B \triangleright D)$.

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

With this new definition in tow we can provide a simple dialetica model for the system LLMS itself, instead of its commutative version LLMS_c. 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 {false, undefined, true}.

DEFINITION 5.2 (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 $(U \overset{\alpha}{\leftarrow\!\!\!+\!\!\!\rightarrow} X)$.
- A morphism from $(U \overset{\alpha}{\leftarrow\!\!\!+\!\!\!\rightarrow} X)$ to $(V \overset{\beta}{\leftarrow\!\!\!+\!\!\!\rightarrow} Y)$ consists a pair of maps, $f\colon U \to V$ and $F\colon Y \to X$, 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 5.3 (structure of $\mathcal{L}$)
Let $A = (U \overset{\alpha}{\leftarrow\!\!\!+\!\!\!\rightarrow} X)$ and $B = (V \overset{\beta}{\leftarrow\!\!\!+\!\!\!\rightarrow} Y)$ be objects of the category $\mathcal{L}$.

- The tensor bifunctor $\otimes\colon \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ given by

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

  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 $I = (\mathbf{1} \overset{\iota}{\leftarrow\!\!\!+\!\!\!\rightarrow} \mathbf{1})$, where ι is the identity relation on $\mathbf{1}$.
- The internal-hom bifunctor $[-,-]_{\mathcal{G}}\colon \mathcal{G}^{op} \times \mathcal{G} \to \mathcal{G}$ is still given by

$$[A, B]_{\mathcal{G}} = (V^U \times X^Y \overset{\beta^\alpha}{\leftarrow\!\!\!+\!\!\!\rightarrow} U \times Y),$$

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

$$V^U \times X^Y \times U \times Y \xrightarrow{\langle \pi_3, \pi_4, eval, eval \rangle} U \times Y \times V \times X \xrightarrow{`\alpha\times\beta'} \mathbf{3} \times \mathbf{3} \xrightarrow{\multimap} \mathbf{3}.$$

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

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

where $(u,v)\alpha \rhd \beta(x,y)$ 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.

- The product bifunctor $\&\colon \mathcal{G}\times\mathcal{G}\to\mathcal{G}$ is given by

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

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 $\top = (\mathbf{1}\overset{e}{\longleftrightarrow}\mathbf{0})$.

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

THEOREM 5.4 (Soundness of $\mathcal{L}$)
The symmetric monoidal closed category $\mathcal{L}$, with bifunctors internal-hom $[-,-]_{\mathcal{L}}$; tensor products $\otimes$ and $\rhd$; and the comonads ! and †, is a model for LLMS. Thus, each entailment $\Gamma\vdash_{\mathsf{LLMS}} A$ corresponds to the existence of a morphism in $\mathcal{L}$, $(f,F)\colon [\![\Gamma]\!]\to[\![A]\!]$.

6 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 realized that we could use Reddy's ideas to give a syntactical characterization to some of the constructions of dialectica categorical models $\mathsf{Dial}_2(Set)$ for Classical Linear Logic. In particular it was interesting to find an intuitive explanation for the comonad † in terms of its relationship to the extra tensor product $\odot$.

We first 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 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. As remarked by a helpful reviewer the work of Bechet *et al.* is related to pomset logic [17] which is also connected to *deep inference* [10, 18]. (In fact pomset logic, under whichever name one prefers, is the only logical system I know of which requires deep inference to be shown cut-free.) Hence interesting connections are expected between the work in this note and deep inference frameworks, but this is left for future work.

Much work remains to be done. Firstly, going back to the original motivation, more work is needed to relate the system LLMS$_c$ to O'Hearn's and Reddy's other work on Algol-like languages[13, 15]. Secondly, from a logical pespective, 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 would be nice if we could find a mathematically significant relationship between linear logic polarization and dialectica categories. Finally it is clear that the work on the inter-relationship between the modalities described here and especially the ones described in [9] are related to the work on sub-exponential modalities in [11], but this relationship needs more investigation.

Acknowledgements

I would like to thank Marcelo da Silva Corrêa for keeping the files that allowed me to produce a version of this work to celebrate E. Hermann Haeusler's birthday.

References

[1] D. Bechet, P. 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 Homology Theory, LNM 80*, pp. 119–140. Springer-Verlag, 1969.

[3] M. Churchill and J. Laird. A logic of sequentiality. In *Computer Science Logic (LectureNotes 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, Pontif´ıcia Univ. Católica PUC-Rio, August 1995. (Electronically available in 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*. Vol. 389 of *LNCS*, pp. 341–356, September 1989.

[9] V. De Paiva. A dialectica model of the lambek calculus. 1991.

[10] A. Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic (TOCL)*, **8**, 1–64, 2007.

[11] V. Nigam and D. Miller. Algorithmic specifications in linear logic with subexponentials. In *Proceedings of the 11th ACM SIGPLAN conference on Principles and practice of declarative programming*, pp. 129–140. ACM, 2009.

[12] P. W. O'Hearn. Linear logic and interference control. In *Categories in Computer Science and Logic*. Vol. 530 of *LNCS*, pp. 74–93, Paris, France, September 1991. Springer-Verlag, Berlin.

[13] P. W. O'Hearn, A. J. Power, M. Takeyama, and R. D. Tennent. Syntactic control of interference revisited. In *MFPS XI*, volume 1 of *Eletronic Notes in Theoretical Computer Science*, New Orleans, March 1995.

[14] U. S. Reddy. A linear logic model of state: preliminar report. Electronic manuscript (ftp),csc.uiuc.edu, directory /pub/reddy/papers, May 1992.

[15] U. S. Reddy. Passivity and independence. In *9th Annual IEEE Symposium on Logic in Computer Science*, pp. 342–352, Paris, France, 1994. IEEE Computer Society Press, 1994.

[16] C. Retoré. *Reseaux et Séquents Ordonnés Mathématiques*. PhD thesis, Université Paris 7, Février 1993.

[17] C. Retoré. Pomset logic: a non-commutative extension of classical linear logic. In *Typed Lambda Calculus and Applications, TLCA'97*; J. R. Hindley and Ph. de Groote, eds, vol. 1210 of LNCS, Springer-Verlag, pp. 300Ð318, 1997.

[18] A. Tiu. A local system for intuitionistic logic. In *Proceedings of LPAR 2006*, vol. 4246 of LNCS, Springer, 2006.

[19] A. Tucker. Categorical modelling of a state oriented version of linear logic. In *Proceedings of the Third ESSLLI Student Session*, Saarbruecken, Germany, August 1998.

Received 28 March 2014