A Dialectica Model of State

Marcelo da Silva Corrêa [1]
e-mail: correa@inf.puc-rio.br

Edward Hermann Haeusler
e-mail: hermann@inf.puc-rio.br

Valeria Correa Vaz de Paiva
Computer Laboratory - University of Cambridge
New Museums Site - Pembroke Street
Cambridge CB2 3QG - U.K.
Computer Laboratory - University of Cambridge
e-mail: Valeria.Paiva@cl.cam.ac.uk

PUC-Rio.MCC21/95

Abstract: Reddy [13] introduced an extended intuitionistic linear calculus to model some features of state-manipulation. His calculus LLMS – for Linear Logic Model of State – includes the connective "before" and an its associated modality †. De Paiva [5] presents a (collection of) dialectica categorical models for Classical Linear Logic, the categories GC. These categories contain an extra tensor product functor $\odot$ and a comonad structure corresponding to a modality related to it. It is surprising that these works arising from completely different motivations can be related in a meaningful way. In this paper we present a dialectica category $\mathcal{G}$ which models the sequent calculus LLMS$_c$, the commutative version of Reddy's LLMS. Furthemore, we present a computational characterization of the fragment $\{\otimes, \odot, \&\}$ of LLMS$_c$ in the framework of CSP [9]. We draw some preliminary conclusions and point out extensions and future work.

Keywords: Intuitionistic Linear Logic, Proof Theory, Categorical Models and Concurrency.

Resumo: Reddy [13] apresenta um cálculo linear intuicionista para modelar alguns aspectos de manipulação de estados. Seu cálculo, chamado "Linear Logic Model of State" (LLMS), inclui o conectivo "before" e uma modalidade †, associada a este conectivo. De Paiva [5] apresenta uma coleção de modelos categóricos dialéticos para a Lógica Linear Clássica em termos das categorias GC. Estas categorias incluem, além do produto tensorial $\otimes$ e a conjunção padrão &, um funtor produto $\odot$ e uma estrutura de comonada para uma modalidade relativa a este produto. Surpreendentemente, estes trabalhos que têm motivações completamente diferentes podem ser relacionados de um modo bastante interessante. Neste trabalho, nós apresentamos uma categoria dialética G que determina um modelo categórico para o cálculo de sequentes LLMS$_c$, uma versão comutativa do cálculo LLMS. Além disso, nós apresentamos uma caracterização computacional do fragmento $\{\otimes, \odot, \&\}$ de LLMS$_c$ em CSP [9]. Nós concluímos o trabalho destacando algumas extensões e trabalhos futuros.

Palavras-Chave: Lógica Linear Intuicionista, Teoria da Prova, Modelos Categóricos e Concorrência.

[1] Also: Dept. de Análise - Univ. Federal Fluminense

1 Introduction

The concept of state-manipulation plays an important role in computation. There are many attempts to formalise this concept, especially in the framework of Girard's Linear Logic ([8, 10, 11, 15, 16, 13]). Some aspects of state-manipulation, like sequencing and regenerative values, are captured in an interesting way by Reddy's work[13]. 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 $\rhd$, called "before", and a "regenerative" storage operator $\dagger$, a modality associated with "before".

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 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 follow this idea: we introduce a new connective $\odot$ which is just the commutative version of Reddy's $\rhd$ and we modify the system LLMS accordingly. Hence the presentation of the system of LLMS$_c$ follows strictly the presentation of LLMS.

Reddy also presents a categorical model for his calculus using a variant of coherent spaces and linear maps. However, if we do consider the commutative version of the connective "before" which is related to interleaving, as we suggest in section 4, we have realised that we end up with a calculus (dubbed LLMS$_c$) for which we had already a dialectica categorical model.

Dialectica models were introduced by de Paiva [3] as a model for the intuitionistic fragment of Linear Logic (the motivation for the model comes from Gödel's Dialectica Interpretation, hence the name). Later, de Paiva introduced the categories GC [4, 5], a different kind of dialectica model, which following a suggestion of Girard, model the whole of Linear Logic. Actually the categories GC have some extra structure (an extra tensor product and an extra modality, associated with this tensor product) and hence can be seen as a model of LLMS$_c$. The extra tensor product, denoted here by $\odot$, can be seen as representing the aleatory combination of elements. The extra (monoidal) comonad can be seen as an extra modality, which we think of as a commutative version of the "regenerative" storage operator.

In the section 2, we present the proof system LLMS$_c$ obtained by modifying LLMS to make the connective "before" commutative. In the section 3, we describe one specific dialectica category (built over Sets) and show that it is a sound model for LLMS$_c$. In the section 4, we present a characterization of the fragment of LLMS$_c$ consisting of cartesian product, the tensor product and the extra tensor product $\odot$ in CSP [9]. Detailed proofs of the results stated here can be found in [2]. Finally, we conclude pointing out some future work.

2 The system LLMS_c

The presentation of the system of LLMS_c follows strictly the presentation of LLMS [13]. 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.*

In order to provide a pomset [2] characterization to a context, the following objects are associated with each context Γ:

- $|\Gamma|$ the set of formula occurrences in Γ, and
- $\geq_\Gamma$, a partial order on $|\Gamma|$;

such that, the pair $(|\Gamma|, \geq_\Gamma)$ is a pomset defined inductively as follows:

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

The Rules of LLMS_c look like those of LLMS [13].

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:

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

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

Additive Rules:

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

[2] partially ordered multiset

Modalities:

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

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

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

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

Next, we present some inferences which are enclosed by the structural rule and which determine interesting properties of the connectives $\otimes$ and $\odot$.

Proposition 2.1 : *The following inferences are allowed by the Structural Rule (***Ser***).*

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

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

$$\textbf{(iv)} \quad \frac{\Delta[(A, B), C] \vdash D}{\Delta[A, (B, C)] \vdash D} \qquad \textbf{(v)} \quad \frac{\Delta[A, (B, C)] \vdash D}{\Delta[(A, B), C] \vdash D}$$

$$\textbf{(vi)} \quad \frac{\Delta[(A; B); C] \vdash D}{\Delta[A; (B; C)] \vdash D} \qquad \textbf{(vii)} \quad \frac{\Delta[A; (B; C)] \vdash D}{\Delta[(A; B); C] \vdash D}$$

$$\textbf{(viii)} \quad \frac{\Delta[(A, B); C] \vdash D}{\Delta[A, (B; C)] \vdash D} \qquad \textbf{(ix)} \quad \frac{\Delta[A; (B, C)] \vdash D}{\Delta[(A; B), C] \vdash D}$$

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

Proof: Regarding that

$$\frac{\Delta[\Gamma'] \vdash \alpha}{\Delta[\Gamma] \vdash \alpha}$$

is the common form of each inference considered here, we must proof in each case that

$$|\Delta[\Gamma]| = |\Delta[\Gamma']| \quad \text{and} \quad \leq_{\Delta[\Gamma]} \subseteq \leq_{\Delta[\Gamma']} .$$

This can be done by means of a structural induction on $\Delta[\Gamma]$, where the base step is the case in which $\Delta[\Gamma]$ is like Γ (and so, $\Delta[\Gamma']$ is like Γ').

Base Step: We have that $|\Gamma| = |\Gamma'|$ for each case **(i)** to **(x)**, since Γ and Γ' have the same occurrences of formulas.

Then, it remains to prove that $\leq_\Gamma \subseteq \leq_{\Gamma'}$.

(i) $\leq_{A,B} = \leq_A \cup \leq_B = \{(A,A),(B,B)\}$

$\leq_{A;B} = \leq_A \cup \leq_B \cup (|A| \times |B|) \cup (|B| \times |A|) = \{(A,A),(B,B),(A,B),(B,A)\}$

Then, $\leq_{A,B} \subseteq \leq_{A;B}$.

The proofs of the cases **(ii)** and **(iii)** are based on the commutativity of $\cup$ and the proofs of the cases **(iv)** and **(v)** are based on the associativity of $\cup$.

$\leq_{A,B} = \{(A,A),(B,B)\} = \leq_{B,A}$

$\leq_{A;B} = \{(A,A),(B,B),(A,B),(B,A)\} = \leq_{B;A}$

$\leq_{A,(B,C)} = \{(A,A),(B,B),(C,C)\} = \leq_{(A,B),C}$

(vi) and (vii) $\leq_{A;(B;C)} = \leq_A \cup \leq_{B;C} \cup (|A| \times |B;C|) \cup (|B;C| \times |A|)$

$\leq_A = \{(A,A)\}$

$\leq_{B;C} = \leq_B \cup \leq_C \cup (|B| \times |C|) \cup (|C| \times |B|) = \{(B,B),(C,C),(B,C),(C,B)\}$

So, $\leq_{A;(B;C)} = \{(A,A),(B,B),(C,C),(B,C),(C,B),(A,B),(A,C),(B,A),(C,A)\}$

$\leq_{(A;B);C} = \leq_{A;B} \cup \leq_C \cup (|A;B| \times |C|) \cup (|C| \times |A;B|)$

$\leq_{A;B} = \{(A,A),(B,B),(A,B),(B,A)\}$

So, $\leq_{(A;B);C} = \{(A,A),(B,B),(A,B),(B,A),(C,C),(A,C),(B,C),(C,A),(C,B)\}$

Then, $\leq_{A;(B;C)} = \leq_{(A;B);C}$

(viii) $\leq_{A,(B;C)} = \leq_A \cup \leq_{B;C} = \{(A,A),(B,B),(C,C),(B,C),(C,B)\}$

$$\begin{aligned}\leq_{(A,B);C} &= \leq_{A,B} \cup \leq_C \cup (|A,B| \times |C|) \cup (|C| \times |A,B|) = \\ &= \{(A,A),(B,B),(C,C),(A,C),(B,C),(C,A),(C,B)\}\end{aligned}$$

Then, $\leq_{A,(B;C)} \subseteq \leq_{(A,B);C}$

(ix) Analogous to the previous case.

(x) $\leq_{(A;B),(C;D)} = \leq_{A;B} \cup \leq_{C;D} =$
$= \{(A,A),(B,B),(A,B),(B,A),(C,C),(D,D),(C,D),(D,C)\}$

$\leq_{(A,C);(B,D)} = \leq_{A,C} \cup \leq_{B,D} \cup (|A,C| \times |B,D|) \cup (|B,D| \times |A,C|)$

$\leq_{A,C} \cup \leq_{B,D} = \{(A,A),(C,C),(B,B),(D,D)\}$

$|A,C| \times |B,D| = \{(A,B),(A,D),(C,B),(C,D)\}$

$|B,D| \times |A,C| = \{(B,A),(B,C),(D,A),(D,C)\}$

Then, $\leq_{(A;B),(C;D)} \subseteq \leq_{(A,C);(B,D)}$

Induction Step: We must consider structural constructions done on the basic contexts analised above. In order to obtain a general treatment, those pairs of basic contexts will be represented by its common form Γ and Γ', respectively.

Suppose that $|\Lambda| = |\Lambda'|$ and $\leq_\Lambda \subseteq \leq_{\Lambda'}$, where contexts Λ and Λ' are constructed on Γ and Γ', respectively.

There are a few cases to consider. Let β be a formula of LLMS$_c$.

(a) $|\Lambda, \beta| = |\Lambda| + |\beta| = |\Lambda'| + |\beta| = |\Lambda', \beta|$

$\leq_{\Lambda,\beta} \subseteq \leq_{\Lambda',\beta}$, since $\leq_{\Lambda,\beta} = \leq_\Lambda \cup \leq_\beta$ and $\leq_{\Lambda',\beta} = \leq_{\Lambda'} \cup \leq_\beta$.

(b) The proof is analogous to a pair of contexts β, Λ and β, Λ' .

(c) $|\Lambda; \beta| = |\Lambda| + |\beta| = |\Lambda'| + |\beta| = |\Lambda'; \beta|$

$\leq_{\Lambda;\beta} = \leq_\Lambda \cup \leq_\beta \cup (|\Lambda| \times |\beta|) \cup (|\beta| \times |\Lambda|)$.

$\leq_{\Lambda',\beta} = \leq_{\Lambda'} \cup \leq_\beta \cup (|\Lambda'| \times |\beta|) \cup (|\beta| \times |\Lambda'|)$.

Then, by induction hypothesis, $\leq_{\Lambda;\beta} \subseteq \leq_{\Lambda';\beta}$.

(d) The proof is analogous to a pair of contexts $\beta; \Lambda$ and $\beta; \Lambda'$.

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

Theorem 2.1 (Cut Elimination) *If a sequent is provable in* LLMS_c*, then it is provable in $LLMS_c$ without an application of the cut rule.*

The proof of this theorem is a consequence of the following lemma and it is made by induction on the number of applications of the cut rule in a proof of a sequent in LLMS_c.

Lemma 2.1 *If π is a proof of a sequent S in* LLMS_c *which contains only one application of the cut rule, occurring as last inference, then there is a cut-free proof of S in* LLMS_c.

Proof: The degree $g(A)$ of a formula A by

$$\begin{array}{rcl} g(A) & = & 1 \quad \text{(for A atomic)} \\ g(A\#B) & = & max(g(A), g(B)) + 1 \\ g(\Diamond A) & = & g(A) + 1 \end{array}$$

The rank of a proof π (ended by a cut rule) is defined as

$$rank(\pi) = rank_l(\pi) + rank_r(\pi),$$

where $rank_l(\pi)$ ($rank_r(\pi)$) is the number of consecutive sequents counting upward from the left (right) upper sequent of the end sequent, that contain the cut formula. Notice that $rank(\pi) \geq 2$.

Considering π be a such proof of the sequent S. Let $\nu(\pi) = \langle n, m \rangle$ be a scale for measuring the complexity of π, where $n = g(A)$ (A is the cut formula) and $m = rank(\pi)$.

Lets consider that

$$\langle n, m \rangle \; < \; \langle n', m' \rangle \text{ iff either } n < n' \text{ or } (n = n' \text{ and } m < m').$$

The proof will be by induction on $\nu(\pi)$ of a proof π, so we will be focusing in the application of the cut rule and its left and right upper sequents, all the time. Special

Cases: The left or right upper sequent is an initial sequent (the cut formula A is atomic, $g(A) = 1$). It includes the cases in which the rank is equal or greater then 2.

$$\dfrac{A \vdash A \quad \begin{matrix}\pi_2 \\ \Gamma[A] \vdash B\end{matrix}}{\Gamma[A] \vdash B} \qquad \dfrac{\begin{matrix}\pi_1 \\ \Gamma \vdash A\end{matrix} \quad A \vdash A}{\Gamma \vdash A}$$

To these proofs there are the following cut-free proofs, respectively:

$$\begin{matrix}\pi_2 \\ \Gamma[A] \vdash B\end{matrix} \qquad \begin{matrix}\pi_1 \\ \Gamma \vdash A\end{matrix}$$

Case 2: the degree of the cut formula is greater than 1.

a) Lets consider a proof π, such that $g(\pi) > 1$ and $rank(\pi) = 2$.

Induction Hypothesis: For every proof π' of a sequent S', which contains a cut rule only as the last inference and $\nu(\pi') < \nu(\pi)$, there is a cut-free proof of S'.

a.1) The cut formula is $A \odot B$

$$\pi = \dfrac{\dfrac{\begin{matrix}\pi_1' \\ \Gamma_1 \vdash A\end{matrix} \quad \begin{matrix}\pi_1'' \\ \Gamma_2 \vdash B\end{matrix}}{\Gamma_1; \Gamma_2 \vdash A \odot B} \qquad \dfrac{\begin{matrix}\pi_2 \\ \Delta[A; B] \vdash C\end{matrix}}{\Delta[A \odot B] \vdash C}}{\Delta[\Gamma_1; \Gamma_2] \vdash C}$$

Consider the following proof π_3

$$\dfrac{\begin{matrix}\pi_1' \\ \Gamma_1 \vdash A\end{matrix} \qquad \begin{matrix}\pi_2 \\ \Delta[A; B] \vdash C\end{matrix}}{\Delta[\Gamma_1; B] \vdash C}$$

$\nu(\pi_3) < \nu(\pi)$, since $g(B) < g(A \odot B)$. So, by induction hypothesis, there is a cut-free proof π_3' of the sequent $\Delta[\Gamma_1; B] \vdash C$.

Then, the following proof also has less complexity than π:

$$\dfrac{\begin{matrix}\pi_1'' \\ \Gamma_2 \vdash B\end{matrix} \qquad \begin{matrix}\pi_3' \\ \Delta[\Gamma_1; B] \vdash C\end{matrix}}{\Delta[\Gamma_1; \Gamma_2] \vdash C}$$

Therefore, by induction hypothesis, there is a cut-free proof of $\Delta[\Gamma_1; \Gamma_2] \vdash C$.

a.2) The cases in which the cut formula is either $A \otimes B$, $A \multimap B$ or $A \& B$ are analogous.

a.3) The cut formula is $!A$

$$\pi = \dfrac{\dfrac{\begin{matrix}\pi_1 \\ \Gamma \vdash A\end{matrix}}{\Gamma \vdash !A} \qquad \dfrac{\dfrac{\begin{matrix}\pi_2 \\ \Delta[A] \vdash C\end{matrix}}{\Delta[\dagger A] \vdash C}}{\Delta[!A] \vdash C}}{\Delta[\Gamma] \vdash C}$$

Then, lets consider the following proof:

$$\dfrac{\begin{array}{cc}\pi_1 & \pi_2 \\ \Gamma \vdash A & \Delta[A] \vdash C\end{array}}{\Delta[\Gamma] \vdash C}$$

Its complexity is less than the complexity of π. Then, by induction hypothesis, there is a cut-free proof of $\Delta[\Gamma] \vdash C$.

We have been considered the case in which it is applied the rule $\dagger$**Der**. Instead of this, if it is applied the rule

$$\frac{\Delta[\epsilon] \vdash C}{\Delta[\dagger A] \vdash C} \dagger \textbf{Weak},$$

we could obtain a direct proof of $\Delta[\Gamma] \vdash C$ from $\Delta[\epsilon] \vdash C$ by some applications of the rules $\dagger$**Weak** and **!Ser**, since Γ contains only ! formulas.

a.4) The case in which $\dagger A$ is the cut formula is analogous.

b)Lets consider a proof π, such that $g(\pi) > 1$ and $rank(\pi) > 2$. So, $rank_l(\pi) > 1$ and/or $rank_r(\pi) > 1$.

Induction Hypothesis: For every proof π' of a sequent S', which contains a cut rule only as the last inference and $\nu(\pi') < \nu(\pi)$, there is a cut-free proof of S'.

b.1) Suppose that $rank_l(\pi) > 1$.

b.1.1) The left upper sequent $\Gamma \vdash A$ is a lower sequent of the structural rule:

$$\pi = \dfrac{\begin{array}{cc}\pi_1 & \\ \dfrac{\Gamma' \vdash A}{\Gamma \vdash A} & \begin{array}{c}\pi_2 \\ \Delta[A] \vdash C\end{array}\end{array}}{\Delta[\Gamma] \vdash C}$$

Consider the following proof π_3 of $\Delta[\Gamma'] \vdash C$.

$$\dfrac{\begin{array}{cc}\pi_1 & \pi_2 \\ \Gamma' \vdash A & \Delta[A] \vdash C\end{array}}{\Delta[\Gamma'] \vdash C}$$

$\nu(\pi_3) < \nu(\pi)$, since que $rank_l(\pi_3) < rank_l(\pi)$. So, by induction hypotesis, there is a cut-free proof π_3' of $\Delta[\Gamma'] \vdash C$.

Moreover, $|\Delta[\Gamma]| = |\Delta[\Gamma']|$ and $\leq_{\Delta[\Gamma]} \subseteq \leq_{\Delta[\Gamma']}$ since Γ and Γ' were used in an application of the structural rule.

Then, the following proof whose its last inference is the structural rule

$$\dfrac{\begin{array}{c}\pi_3' \\ \Delta[\Gamma'] \vdash C\end{array}}{\Delta[\Gamma] \vdash C}$$

is a cut-free proof of $\Delta[\Gamma] \vdash C$.

b.1.2) The left upper sequent is a lower sequent of an application of a rule whose principal formula is not the cut formula.

$$\pi = \cfrac{\cfrac{\begin{array}{c}\pi_1\\ \Gamma[B_1; B_2] \vdash A\end{array}}{\Gamma[B_1 \odot B_2] \vdash A} \qquad \begin{array}{c}\pi_2\\ \Delta[A] \vdash C\end{array}}{\Delta[\Gamma[B_1 \odot B_2]] \vdash C}$$

Consider the following proof π_3 of $\Delta[\Gamma[B_1; B_2]] \vdash C$.

$$\cfrac{\begin{array}{c}\pi_1\\ \Gamma[B_1; B_2] \vdash A\end{array} \qquad \begin{array}{c}\pi_2\\ \Delta[A] \vdash C\end{array}}{\Delta[\Gamma[B_1; B_2]] \vdash C}$$

Analogously to the previous case, π_3 has less complexity than π. So, by induction hypotesis, there is a cut-free proof π_3' of $\Delta[\Gamma[B_1; B_2]] \vdash C$.

Then, the following proof whose its last inference is $\odot\mathcal{L}$

$$\cfrac{\begin{array}{c}\pi_3'\\ \Delta[\Gamma[B_1; B_2]] \vdash C\end{array}}{\Delta[\Gamma[B_1 \odot B_2]] \vdash C}$$

is a cut-free proof of $\Delta[\Gamma[B_1 \odot B_2]] \vdash C$.

The cases in which the rule applied is either $\otimes\mathcal{L}$, $\mathbf{1}\mathcal{L}$, $\multimap\mathcal{L}$, $1\&\mathcal{L}$, $2\&\mathcal{L}$, †**Weak**, †**Der**, †**Thread**, **!Ser** or **!Contr** are analogous.

b.2) Suppose that $rank_r(\pi) > 1$.

b.2.1) The right upper sequent is a lower sequent of the structural rule. There are many cases to consider, especially those proved in the Proposition 2.1:

$$\frac{\Delta[B_1; B_2] \vdash C}{\Delta[B_1, B_2] \vdash C} \qquad \frac{\Delta[B_2, B_1] \vdash C}{\Delta[B_1, B_2] \vdash C} \qquad \frac{\Delta[B_2; B_1] \vdash C}{\Delta[B_1; B_2] \vdash C}$$

$$\frac{\Delta[(B_1, B_2), B_3] \vdash C}{\Delta[B_1, (B_2, B_3)] \vdash C} \qquad \frac{\Delta[B_1, (B_2, B_3)] \vdash C}{\Delta[(B_1, B_2), B_3] \vdash C} \qquad \frac{\Delta[(B_1; B_2); B_3] \vdash C}{\Delta[B_1; (B_2; B_3)] \vdash C} \qquad \frac{\Delta[B_1; (B_2; B_3)] \vdash C}{\Delta[(B_1; B_2); B_3] \vdash C}$$

$$\frac{\Delta[(B_1, B_2); B_3] \vdash C}{\Delta[B_1, (B_2; B_3)] \vdash C} \qquad \frac{\Delta[B_1; (B_2, B_3)] \vdash C}{\Delta[(B_1; B_2), B_3] \vdash C}$$

$$\frac{\Delta[(B_1, B_3); (B_2, B_4)] \vdash C}{\Delta[(B_1; B_2), (B_3; B_4)] \vdash C}$$

where the cut formula can be any B_i. We will prove only the first one and the others can be proved in an analogous way.

$$\pi = \cfrac{\begin{array}{c}\pi_1\\ \Gamma \vdash A\end{array} \qquad \cfrac{\begin{array}{c}\pi_2\\ \Delta[A; B] \vdash C\end{array}}{\Delta[A, B] \vdash C}}{\Delta[\Gamma, B] \vdash C}$$

Consider the following proof π_3 of $\Delta[\Gamma; B] \vdash C$.

$$\frac{\begin{array}{c}\pi_1\\ \Gamma \vdash A\end{array} \qquad \begin{array}{c}\pi_2\\ \Delta[A; B] \vdash C\end{array}}{\Delta[\Gamma; B] \vdash C}$$

π_3 has less complexity than π. So, by induction hypotesis, there is a cut-free proof π_3' of $\Delta[\Gamma; B] \vdash C$. Moreover, $|\Delta[\Gamma; B]| = |\Delta[\Gamma, B]|$ and $\leq_{\Delta[\Gamma;B]} \subseteq \leq_{\Delta[\Gamma,B]}$ since $\Delta[A; B]$ and $\Delta[A, B]$ were used in an application of the structural rule.

Then, the following proof whose its last inference is the structural rule

$$\frac{\begin{array}{c}\pi_3'\\ \Delta[\Gamma; B] \vdash C\end{array}}{\Delta[\Gamma, B] \vdash C}$$

is a cut-free proof of $\Delta[\Gamma] \vdash C$.

b.2.2) The right upper sequent is a lower sequent of a rule whose principal formula is not the cut formula.

The cases enclosed here are analogous to the case proved in (b.1.2).

b.2.3) The right upper sequent is a lower sequent of a rule whose principal formula is the cut one.

b.2.3.1) Cut formula $!A$

$$\pi = \frac{\dfrac{\begin{array}{c}\pi_1\\ \Gamma \vdash A\end{array}}{\Gamma \vdash !A} \qquad \dfrac{\begin{array}{c}\pi_2\\ \Delta[!A, !A] \vdash C\end{array}}{\Delta[!A] \vdash C}}{\Delta[\Gamma] \vdash C}$$

where Γ is an independent context with only ! formulas.

The complexity of the following proof is less than the complexity of π.

$$\frac{\dfrac{\begin{array}{c}\pi_1\\ \Gamma \vdash A\end{array}}{\Gamma \vdash !A} \qquad \begin{array}{c}\pi_2\\ \Delta[!A, !A] \vdash C\end{array}}{\Delta[\Gamma, !A] \vdash C}$$

Then, there is cut-free proof π_3' of $\Delta[\Gamma, !A] \vdash C$. So, the following proof also has less complexity than π.

$$\frac{\dfrac{\begin{array}{c}\pi_1\\ \Gamma \vdash A\end{array}}{\Gamma \vdash !A} \qquad \begin{array}{c}\pi_3'\\ \Delta[\Gamma, !A] \vdash C\end{array}}{\Delta[\Gamma, \Gamma] \vdash C}$$

Then, there is a cut-free proof π_4' of $\Delta[\Gamma, \Gamma] \vdash C$.

In the other hand, Γ has only occurrences of "," operator and ! formulas. So, there is a cut-free proof of $\Delta[\Gamma] \vdash C$ from π_4' by some permutations allowed by the Structural Rule (Proposition 2.1)

$$\mathcal{I}1 = \frac{\Lambda[B_2, B_1] \vdash C}{\Lambda[B_1, B_2] \vdash C}$$

and contractions (**!Contr**).

$$\cfrac{\cfrac{\begin{array}{c}\pi_4' \\ \Delta[\Gamma, \Gamma] \vdash C\end{array}}{\begin{array}{c}\cdots \\ (\mathcal{I}1 \text{ and } \textbf{!Contr}) \\ \cdots\end{array}}}{\Delta[\Gamma] \vdash C}$$

b.2.3.2) Cut formula $\dagger A$

$$\cfrac{\cfrac{\begin{array}{c}\pi_1 \\ \Gamma \vdash A\end{array}}{\Gamma \vdash \dagger A} \quad \cfrac{\begin{array}{c}\pi_2 \\ \Delta[\dagger A; \dagger A] \vdash C\end{array}}{\Delta[\dagger A] \vdash C}}{\Delta[\Gamma] \vdash C}$$

Analogously to the previous case, there is a cut-free proof π_4' of $\Delta[\Gamma; \Gamma] \vdash C$. Although, Γ has only occurrences of "," operator and ! or $\dagger$ formulas.

Lets consider $\Gamma' \equiv (!\gamma_1, \cdots, !\gamma_{i-1}), (\dagger\gamma_i, \cdots, \dagger\gamma_n)$, where every $!\gamma_j$ and $\dagger\gamma_k$ are occurrences of formulas in Γ, $1 \leq j \leq i-1$ and $i \leq k \leq n$; and the following inferences allowed by the Strucutural Rule (Proposition 2.1) in addition to $\mathcal{I}1$.

$$\mathcal{I}2 = \frac{\Lambda[B_1; B_2] \vdash C}{\Lambda[B_1, B_2] \vdash C}$$

$$\mathcal{I}3 = \frac{\Lambda[(B_1, B_3); (B_2, B_4)] \vdash C}{\Lambda[(B_1; B_2), (B_3; B_4)] \vdash C}$$

and contractions (**!Contr**).

Then, there is the following cut-free proof of $\Delta[\Gamma] \vdash C$:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\pi'_4}{\Delta[\Gamma;\Gamma] \vdash C}}{\cdots(\mathcal{I}1)\cdots}}{\Delta[\Gamma';\Gamma'] \vdash C}}{\cdots(\mathcal{I}3)\cdots}}{\Delta[(!\gamma_1;!\gamma_1),\cdots,(!\gamma_{i-1};!\gamma_{i-1}),(\dagger\gamma_i;\dagger\gamma_i),\cdots,(\dagger\gamma_n;\dagger\gamma_n)] \vdash C}}{\cdots(\dagger\mathbf{Thread})\cdots}
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\cdots(\dagger\mathbf{Thread})\cdots}{\Delta[(!\gamma_1;!\gamma_1),\cdots,(!\gamma_{i-1};!\gamma_{i-1})),\dagger\gamma_i,\cdots,\dagger\gamma_n] \vdash C}}{\cdots(\mathcal{I}2)\cdots}}{\Delta[(!\gamma_1,!\gamma_1),\cdots,(!\gamma_{i-1},!\gamma_{i-1}),\dagger\gamma_i,\cdots,\dagger\gamma_n] \vdash C}}{\cdots(\mathbf{!Contr})\cdots}}{\Delta[!\gamma_1,\cdots,!\gamma_{i-1},\dagger\gamma_i,\cdots,\dagger\gamma_n] \vdash C}}{\dfrac{\cdots(\mathcal{I}1)\cdots}{\Delta[\Gamma] \vdash C}}
$$

b.2.3.3) Cases in which the formula has two different occurrences in the right upper sequent: as the cut formula and as the principal formula

$$
\pi = \dfrac{\begin{array}{c}\pi_1 \\ \Gamma \vdash A \odot B\end{array} \qquad \dfrac{\begin{array}{c}\pi_2 \\ (\Delta[A \odot B])[A;B] \vdash C\end{array}}{(\Delta[A \odot B])[A \odot B] \vdash C}}{(\Delta[\Gamma])[A \odot B] \vdash C}
$$

Consider the following proof π_3 of $(\Delta[\Gamma])[A;B]] \vdash C$.

$$
\dfrac{\begin{array}{c}\pi_1 \\ \Gamma \vdash A \odot B\end{array} \qquad \begin{array}{c}\pi_2 \\ (\Delta[A \odot B])[A;B] \vdash C\end{array}}{(\Delta[\Gamma])[A;B] \vdash C}
$$

π_3 has less complexity than π. So, by induction hypotesis, there is a cut-free proof π'_3 of $(\Delta[\Gamma])[A;B] \vdash C$.

Then, the following proof is a cut-free proof of $(\Delta[\Gamma)[A \odot B] \vdash C$.

$$
\dfrac{\begin{array}{c}\pi'_3 \\ (\Delta[\Gamma])[A;B] \vdash C\end{array}}{(\Delta[\Gamma])[A \odot B] \vdash C} \odot\mathcal{L}
$$

The cases in which the rule applied is either $\otimes\mathcal{L}$, $-\!\circ\mathcal{L}$, $1\&\mathcal{L}$, $2\&\mathcal{L}$, $\dagger$**Weak**, $\dagger$**Der**, $\dagger$**Thread**, **!Ser** or **!Contr** are analogous.

Notice that, there are a few cases in which the principal formula occurs in the right side of the right upper sequent, but they can be reduced to initial sequent.

3 A Categorical Model for LLMS_c

To present a categorical characterization of LLMS_c, we consider an instantiation $\mathcal{G}$ of the (symmetric monoidal closed) categories GC, developed in [5]. 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 $\mathcal{G}$.

In the most general case a GC-like category is obtained from a symmetric monoidal closed category C with products, (and satisfying some other conditions [7]) but for purposes of this paper, our base category will be **Sets**. Readers interested in the (several already known) generalizations of this work should consult [7, 6].

Recall that the partial-order with two elements **2**, where one should think of 0 as false and 1 as true, is a closed-poset. It has a tensor product ($\otimes$) and an internal-hom ($\multimap$) such that $a \otimes b \leq c$ iff $a \leq b \multimap c$. Actually $\otimes$ is the usual conjunction $\wedge$ and the internal-hom *linear* is the usual Heyting implication $\rightarrow$; these operations can be described by the usual truth-tables.

Definition 3.1 : *The category $\mathcal{G}$ is described by:*
- *Objects are relations on* **Sets**, *that is maps $U \times X \xrightarrow{\alpha} 2$, written as $(U \overset{\alpha}{\longleftrightarrow} X)$.*
- *A morphism from $(U \overset{\alpha}{\longleftrightarrow} X)$ to $(V \overset{\beta}{\longleftrightarrow} Y)$ consists a pair of maps, $f\colon U \rightarrow V$ and $F\colon Y \rightarrow X$, 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 $f(u)\beta y$. This could represented by the following diagram

$$\begin{tikzcd}
U \times Y \arrow[r, "\langle \pi_1, F\rangle"] \arrow[d, "f \times id_Y"'] \arrow[dr, phantom, "\geq"] & U \times X \arrow[d, "\alpha"] \\
V \times Y \arrow[r, "\beta"'] & 2
\end{tikzcd}$$

We usually depict a morphism (f, F) in $\mathcal{G}$ as follows:

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

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

Definition 3.2 : *Let* $A = (U \xleftarrow{\alpha} X)$ *and* $B = (V \xleftarrow{\beta} Y)$ *be objects of* $\mathcal{G}$.

- *The tensor bifunctor* $\otimes : \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ *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* ι *is the identity relation on* $\mathbf{1}$.

- *The tensor bifunctor* $\odot : \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ *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* $[-,-]_{\mathcal{G}} : \mathcal{G}^{op} \times \mathcal{G} \to \mathcal{G}$ *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* β^α *is given by the composition*

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

- *The product bifunctor* $\& : \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ *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* $\top = (\mathbf{1} \xleftarrow{e} \mathbf{0})$.

We could also describe coproducts ($\oplus$) and a par connective ($\Box$), but these will not play an important role in the following discussion.

Lemma 3.1 : *For any* A *and* B *objects of* $\mathcal{G}$ *there are natural morphisms:*

(i) $\tau_{(A,B)} : A \otimes B \to A \odot B$[3]

(ii) *weak-distributivity:*

$$w_{(A,B,C)} : A \otimes (B \odot C) \to (A \otimes B) \odot C$$

(iii) $v_{(A,B,C,D)} : (A \odot B) \otimes (C \odot D) \to (A \otimes C) \odot (B \otimes D)$

Proof:

(i) The natural morphism $\tau_{(A,B)}$ is given by $(id_{U \times V}, (k_1, k_2))$,

$$\begin{tikzcd}
U \times V \arrow[d, "id_{U\times V}"'] & X^V \times Y^U \arrow[l, "\alpha \otimes \beta"'] \\
U \times V & X \times Y \arrow[l, "\alpha \odot \beta"] \arrow[u, "(k_1, k_2)"']
\end{tikzcd}$$

where $(k_1, k_2) : X \times Y \to X^V \times Y^U$ consists of constant maps – or exponential transposes of the projections – in each coordinate.

[3] There is another natural morphism $\sigma_{A,B} : (A \odot B) \to (A \Box B)$, which shall not concern us here.

(ii) The natural morphism $w_{(A,B,C)}$ corresponding to the first weak-distributivity is given by $(\alpha, (k_1, k_2, k_3))$,

$$\begin{tikzcd}
U \times (V \times W) \arrow[d, "\alpha"'] & X^{V \times W} \times (Y \times Z)^U \arrow[l, "\alpha \otimes (\beta \odot \gamma)"'] \\
(U \times V) \times W & (X^V \times Y^U) \times Z \arrow[l, "(\alpha \otimes \beta) \odot \gamma"] \arrow[u, "(k_1, k_2, k_3)"']
\end{tikzcd}$$

where $\alpha: U \times (V \times W) \to (U \times V) \times W$ is the associativity isomorphism; $k_1: X^V \to X^{V \times W} \cong (X^V)^W$ is the constant map; $k_2: Y^U \to Y^U$ is the identity map and $k_3: Z \to Z^U$ is the constant map again (note that we need more associativity and the isomorphism $(Y \times Z)^U \cong Y^U \times Z^U$ here).

(iii) The natural morphism $v_{(A,B,C,D)}$ is given by $(\gamma, (k_1, k_2, k_3, k_4))$,

$$\begin{tikzcd}
(U \times V) \times (W \times P) \arrow[d, "\gamma"'] & (X \times Y)^{W \times P} \times (Z \times Q)^{U \times V} \arrow[l, "(\alpha \odot \beta) \otimes (\gamma \odot \rho)"'] \\
(U \times W) \times (V \times P) & (X^W \times Z^U) \times (Y^P \times Q^V) \arrow[l, "(\alpha \otimes \gamma) \odot (\beta \otimes \rho)"] \arrow[u, "(k_1, k_2, k_3, k_4)"']
\end{tikzcd}$$

where
$\gamma: (U \times V) \times (W \times P) \to (U \times W) \times (V \times P)$ is the commutativity isomorphism and $k_1: X^W \to X^{W \times P}$ $k_2: Z^U \longrightarrow Z^{(U \times V)}$; $k_3: Y^P \to Y^{W \times P}$ and $k_4: Q^V \to Q^{U \times V}$ are constant maps and we use some isomorphisms to get the parenthesis in the right places.

Recall that **Sets** does have free (commutative) monoid structures.

- There is an adjunction $F \dashv \mathrm{U}: \mathbf{Mon}_c \to \mathbf{Sets}$ where $\mathbf{Mon}_c$ is the category of commutative monoids, U is the forgetful functor and in

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

  X^* can be viewed as finite sequences (up to permutation) of elements of X, $e_X: \mathbf{1} \to X^*$ is the empty sequence of elements of X and $m_X: X^* \times X^* \to X^*$ means concatenation of 'sequences'.

- There is a monad in **Sets**, $(*, \eta_*, \mu_*)$, where $* = U \circ F: \mathbf{Sets} \to \mathbf{Sets}$, and we write f^* for f applied to each element of the sequence. This monad's unit is given by the singleton sequence $(\eta_*)_X: X \to X^*$ and its multiplication $(\mu_*)_X: X^{**} \to X^*$ by forgetting parenthesis.

We will need to consider another monad structure in **Sets**. 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 3.3 :
Consider the following endofunctors on $\mathcal{G}$:

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

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

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

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

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

   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 $U \times X^* \xrightarrow{\dagger\alpha} 2$ which is the composition

$$U \times X^* \xrightarrow{m'_{U,X}} (U \times X)^* \xrightarrow{\alpha^*} 2^* \longrightarrow 2$$

(the map $m'_{U,X}: U \times X^* \to (U \times X)^*$ 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 $2^* \to 2$ using the tensor product in 2) and the relation $G\alpha$ is given by the composition

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

Lemma 3.2 : *The functors $\dagger$ and G have natural 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 $X \overset{(\eta_*)_X}{\to} X^*$ and the comultiplication $(\delta_\dagger)_A: \dagger A \to \dagger^2 A$ is given by identity on U and forgetting brackets $X^{**} \overset{(\mu_*)_X}{\to} X^*$ in the second coordinate, as in the diagram

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

Moreover this comonad is *monoidal* with respect to the tensor structure $\odot$, that is there is a natural morphism $\dagger A \odot \dagger B \to \dagger(A \odot B)$ given by

$$\begin{tikzcd}
U \times V \arrow[d, "id_{U\times V}"'] & X^* \times Y^* \arrow[l, "\dagger\alpha \odot \dagger\beta"', "|" marking] \\
U \times V & (X \times Y)^* \arrow[l, "\dagger(\alpha \odot \beta)", "|" marking] \arrow[u, "r_{(X,Y)}"']
\end{tikzcd}$$

where $r_{(X,Y)}: (X \times Y)^* \to X^* \times Y^*$ transforms a sequence of pairs $\langle x_i, y_i\rangle$ into the the pair of sequences $\langle x_i\rangle$ and $\langle y_i\rangle$.

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

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

This comonad is monoidal with respect to the tensor structure $\otimes$, as there is a natural morphism $n_{A,B}: GA \otimes GB \to G(A \otimes B)$ given by

$$\begin{tikzcd}
U \times V \arrow[d] & (X^U)^V \times (Y^V)^U \arrow[l, "G\alpha \otimes G\beta"', "|" marking] \\
U \times V & (X^V \times Y^U)^{U\times V} \arrow[l, "G(\alpha \otimes \beta)", "|" marking] \arrow[u, "r'"']
\end{tikzcd}$$

where $r': X^{U\times V\times V} \times Y^{U\times U\times V} \to X^{U\times V} \times Y^{U\times V}$ is just restriction to the respective diagonal.

Now there is a distributive law relating our comonads $\dagger$ and G. Then, by abstract nonsense [1] we have that these comonads compose and that G lifts to the category of coalgebras for $\dagger$. The next lemma collects some facts about G and $\dagger$.

Lemma 3.3 :

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

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

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*

$$(X^U)^* \times U \xrightarrow{m'_{X^U,U}} (X^U \times U)^* \xrightarrow{ev^*} X^*$$

- *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*

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

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".*

- *There is a comonad morphism* $\kappa\colon ! \to \dagger$*, where* $\kappa_A\colon !A \to \dagger A$ *is given by*

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

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

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

Definition 3.4 : *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 long composition

$$(((X^*)^U)^*)^U \times U \overset{\langle ev,\pi\rangle}{\longrightarrow} ((X^*)^U)^* \times U \overset{\lambda\times U}{\longrightarrow} (X^{**})^U \times U \overset{ev}{\to} X^{**} \overset{(\mu_*)_X}{\to} X^*$$

We emphasize some properties which are used in the proof of soundness of the model.

Proposition 3.1 :

1. *For any $\mathcal{G}$-object A, $!A$ has a comonoid structure $(!A, d_A, e_A)$ with respect to $\otimes$.*

   The comonoid counit $e_A\colon !A \to I$ is given by the terminal map on U and the map $\mathbf{1} \to (X^*)^U$.

   The comultiplication $d_A\colon !A \to !A \otimes !A$ is given by the diagonal $\Delta\colon U \to U \times U$, and the map $\theta\colon (X^*)^{U\times U} \times (X^*)^{U\times U} \to (X^*)^U$. The map θ is given by taking a pair of functions (ϕ,φ), each of them of the form $U \times U \to X^*$, to the following map

   $$U \overset{\Delta}{\to} U \times U \overset{<\phi,\varphi>}{\longrightarrow} X^* \times X^* \overset{m_X}{\to} X^*$$

   where $m_X\colon X^* \times X^* \to X^*$ is concatenation of sequences.

   This comonoid structure determines a functor $R_!\colon \mathcal{G} \to \mathbf{Comon}_{\otimes}\mathcal{G}$ which takes A to $(!A, d_A, e_A)$.

2. *For any $\mathcal{G}$-object A, $\dagger A$ has a natural comonoid structure with respect to $\odot$. 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 $\mathrm{U}\colon \mathbf{Comon}_{\odot}\mathcal{G} \to \mathcal{G}$.*

3. *There are natural isomorphisms for all objects A and B in $\mathcal{G}$*

   $$!(A\&B) \cong !A \otimes !B$$

Lemma 3.4 : *If there is a morphism from* $[\![\Gamma[B]\,]\!]$ *to* $[\![C]\!]$ *and a morphism from* $[\![A]\!]$ *to* $[\![B]\!]$*, then there is a morphism from* $[\![\Gamma[A]\,]\!]$ *to* $[\![C]\!]$*.*

Proof: Suppose that there are morphisms $(f, F)\colon [\![\Gamma[B]\,]\!] \to [\![C]\!]$ and $(h, H)\colon [\![A]\!] \to [\![B]\!]$. Let us consider the following cases:

(i) $\Gamma[B]$ has only the formula B. Then the morphism is $(f, F) \circ (h, H)$.

(ii) $\Gamma[B]$ has the form $(B, \Delta_1), \Delta_2$. Then $[\![\Gamma[B]\,]\!] = ([\![B]\!] \otimes [\![\Delta_1]\!]) \otimes [\![\Delta_2]\!]$ So, there is a morphism $(f, F) \circ ((h, H) \otimes ID_{[\![\Delta_1]\!]}) \otimes ID_{[\![\Delta_2]\!]}$ from $[\![\Gamma[A]\,]\!]$ to $[\![C]\!]$.

The permutations of this possibility are proved analogously.

(iii) $\Gamma[B]$ has the form $(B; \Delta_1), \Delta_2$. So, $[\![\Gamma[B]\,]\!] = ([\![B]\!] \odot [\![\Delta_1]\!]) \otimes [\![\Delta_2]\!]$ So, there is a morphism $(f, F) \circ ((h, H) \odot ID_{[\![\Delta_1]\!]}) \otimes ID_{[\![\Delta_2]\!]}$ from $[\![\Gamma[A]\,]\!]$ to $[\![C]\!]$.

The permutations of this possibility are proved analogously.

(iv) The case in which $\Gamma[B]$ has either the form $(B, \Delta_1); \Delta_2$ or the form $(B; \Delta_1); \Delta_2$ and all permutations of these possibilities are analogous to the second and third cases.

(v) The cases in which $\Gamma[B]$ contains two occurrences of formulas are particular cases of the above ones.

Theorem 3.1 (Soundness) : *The symmetric monoidal closed category* $\mathcal{G}$*, with bifunctors internal hom* $[-, -]_{\mathcal{G}}$*; tensor product* $\otimes$*; product* $\odot$*; cartesian product* & *and the comonads* ! *and* $\dagger$*, is a model for* LLMS$_c$*. Thus, each entailment* $\Gamma \vdash_{\mathrm{LLMS}_c} A$ *corresponds to the existence of a morphism in* $\mathcal{G}$*,* $(f, F)\colon [\![\Gamma]\!] \to [\![A]\!]$*.*

Proof: We have only to check that the rules of LLMS$_c$ are preserved by the interpretation. First, lets consider that to each context Γ, $[\![\Gamma]\!] = [\![\gamma]\!]$, where γ is a formula obtained from Γ by substituting any ocurrences of the operators "," and ";" by the connectives $\otimes$ and $\odot$, respectively.

(i) *Structural Rule*

The proofs of the main inferences allowed by the Structural Rule (Proposition 2.1) are based on the Lemma 3.4, the Lemma 3.1 and on the facts that the functors $\otimes$ and $\odot$ are commutative and associative.

(ii) *Identity Rules*

- To each initial sequent inferred by the rule **Id** corresponds an identity morphism in $\mathcal{G}$.
- The proof of the **Cut** rule can be obtained directly from the Lemma 3.4.

(iii) *Multiplicative Rules*

- The rules $\otimes\mathcal{R}$ and $\odot\mathcal{R}$ only say that the tensor product $\otimes$ and the product $\odot$ are bifunctors:
  If $[\![\Gamma]\!] \overset{(f,F)}{\rightarrow} [\![A]\!]$ and $[\![\Delta]\!] \overset{(h,H)}{\rightarrow} [\![B]\!]$ then $(f,F)\otimes(h,H): [\![\Gamma\otimes\Delta]\!] \rightarrow [\![A\otimes B]\!]$.
  If $[\![\Gamma]\!] \overset{(f,F)}{\rightarrow} [\![A]\!]$ and $[\![\Delta]\!] \overset{(h,H)}{\rightarrow} [\![B]\!]$ then $(f,F)\odot(h,H): [\![\Gamma\odot\Delta]\!] \rightarrow [\![A\odot B]\!]$.
- The proofs of the rules $\otimes\mathcal{L}$ and $\odot\mathcal{L}$ follow immediately by the Lemma 3.4, since $[\![A,B]\!] = [\![A\otimes B]\!]$ and $[\![A;B]\!] = [\![A\odot B]\!]$.
- The proofs of the rules $\mathbf{1}\mathcal{R}$ and $\mathbf{1}\mathcal{L}$ are based on the fact that $[\![\mathbf{1}]\!] = I = \mathbf{1} \overset{\iota}{\leftarrow} \mathbf{1}$ is the unit of the bifunctors $\otimes$ and $\odot$.
- The rules $\multimap\mathcal{R}$ and $\multimap\mathcal{L}$ reflect the monoidal closed structure. Thus,

  If $[\![\Gamma\otimes A]\!] \overset{(f,F)}{\longrightarrow} [\![B]\!]$, using the adjunction we get $[\![\Gamma]\!] \overset{\overline{(f,F)}}{\longrightarrow} [\,[\![A]\!],[\![B]\!]\,]_{\mathcal{G}}$, where $[\,[\![A]\!],[\![B]\!]\,]_{\mathcal{G}} = [\![A\multimap B]\!]$ and $[\![\Gamma\otimes A]\!] = [\![\Gamma]\!]\otimes[\![A]\!]$.

  Further, if $[\![\Gamma]\!] \overset{(f,F)}{\longrightarrow} [\![A]\!]$, then we have
  $[\![\Gamma]\!]\otimes[\![A\multimap B]\!] \overset{(f,F)\times Id}{\longrightarrow} [\![A]\!]\otimes[\![A\multimap B]\!] \overset{ev}{\rightarrow} [\![B]\!]$.
  So, by the Lemma 3.4, if there is a morphism from $[\![\Gamma]\!]$ to $[\![A]\!]$ and there is a morphism from $[\![\Delta[B]\,]\!]$ to $[\![C]\!]$, then there is a morphism from $[\![\Delta[\Gamma, A\multimap B]\,]\!]$ to $[\![C]\!]$.

(iv) *Additive Rules*

The additive rules characterize the properties of a cartesian product, where $[\![A\&B]\!] = [\![A]\!]\&[\![B]\!]$. Thus,

- $[\![\top]\!] = (\mathbf{1} \overset{e}{\longleftarrow\!\!\!+} \mathbf{0})$ is the unit of the bifunctor $\&$.
- If $[\![\Gamma]\!] \overset{(f,F)}{\longrightarrow} [\![A]\!]$ and $[\![\Gamma]\!] \overset{(h,H)}{\longrightarrow} [\![B]\!]$ then $((f,h), \begin{pmatrix} F \\ H \end{pmatrix}): [\![\Gamma]\!] \rightarrow [\![A\&B]\!]$.
- The rules $\mathbf{1}\&\mathcal{L}$ and $\mathbf{2}\&\mathcal{L}$ correspond to the existence of projections in $\mathcal{G}$, $p_1: [\![A]\!]\&[\![B]\!] \rightarrow [\![A]\!]$ and $p_2: [\![A]\!]\&[\![B]\!] \rightarrow [\![B]\!]$ given by $p_1 = (\pi_1, j_1)$ and $p_2 = (\pi_2, j_2)$, where $\pi_1: U\times V \rightarrow U$ and $\pi_2: U\times V \rightarrow V$ are canonical projections and $\jmath_1: X \rightarrow X+Y$ and $\jmath_2: Y \rightarrow X+Y$ are canonical injections.

(v) *Modalities*

The proofs of the rules of the modalities $\dagger$ and $!$ are based on its comonad properties described in the Lemma 3.2, Lemma 3.3, Proposition 3.1 and using the Lemma 3.4, where $[\![\dagger A]\!] = \dagger[\![A]\!]$ and $[\![!A]\!] = ![\![A]\!]$.

- To prove the rules $\dagger$**Weak**, $\dagger$**Der**, $\dagger$**Thread** it is enough to consider the maps

$$e'_{[\![A]\!]}: \dagger[\![A]\!] \rightarrow I, \quad (\epsilon_\dagger)_{[\![A]\!]}: \dagger[\![A]\!] \rightarrow [\![A]\!] \quad \text{and} \quad d'_{[\![A]\!]}: \dagger[\![A]\!] \rightarrow \dagger[\![A]\!]\odot\dagger[\![A]\!]$$

- To prove the rules **!Contr** and **!Ser** it is enough to consider the maps

$$d_{[\![A]\!]}: ![\![A]\!] \rightarrow ![\![A]\!]\otimes![\![A]\!] \quad \text{and} \quad \kappa_{[\![A]\!]}: ![\![A]\!] \rightarrow \dagger[\![A]\!]$$

- To prove the rule

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

suppose that there is a morphism $h_1 \colon [\![\Gamma]\!] \to [\![A]\!]$. First, notice that, Γ corresponds to $!\gamma_1, \cdots, !\gamma_i, \dagger\gamma_{i+1}, \cdots, \dagger\gamma_n$.

For any $\mathcal{G}$-object A there are the following maps:

$$!A \stackrel{(\delta_!)_A}{\to} !!A \stackrel{\kappa_{!A}}{\to} \dagger !A \quad \text{and} \quad \dagger\, A \stackrel{(\delta_\dagger)_A}{\to} \dagger\dagger\, A$$

So, there is the following map

$$h_2 \colon [\![\Gamma]\!] \to \dagger ![\![\gamma_1]\!] \otimes \cdots \otimes \dagger ![\![\gamma_i]\!] \otimes \dagger\dagger\, [\![\gamma_{i+1}]\!] \otimes \cdots \otimes \dagger\dagger\, [\![\gamma_n]\!]$$

Using (a generalisation of) the maps $m_{A,B}$ of Lemma 3.3 we obtain a map

$$h_3 \colon \dagger ![\![\gamma_1]\!] \otimes \cdots \otimes \dagger ![\![\gamma_i]\!] \otimes \dagger\dagger\, [\![\gamma_{i+1}]\!] \otimes \cdots \otimes \dagger\dagger\, [\![\gamma_n]\!] \to \dagger [\![\Gamma]\!]$$

there is a map

$$\dagger h_1 \colon \dagger [\![\Gamma]\!] \to \dagger [\![A]\!]$$

Therefore, there is a map $\dagger h_1 \circ (h_3 \circ h_2)$ from $[\![\Gamma]\!]$ to $\dagger [\![A]\!] = [\![\dagger A]\!]$.

- To prove the rule

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

we first recall that Γ corresponds to $!\gamma_1, \cdots, !\gamma_n$. As we have seen in the proposition above, there is a isomorphism

$$[\![!\gamma_1, \cdots, !\gamma_n]\!] = ![\![\gamma_1]\!] \otimes \cdots \otimes ![\![\gamma_n]\!] \cong !([\![\gamma_1]\!] \& \cdots \& [\![\gamma_n]\!]) = !H$$

Given any map $!H \to [\![A]\!]$ we can obtain a map $!H \stackrel{\delta_!}{\longrightarrow} !!H \stackrel{!(f,F)}{\longrightarrow} ![\![A]\!]$.

4 A fragment of LLMS_c in CSP

In this section, we present an interpretation of a fragment of LLMS_c including the operators tensor product $\otimes$, cartesian product $\&$ and the operator $\odot$ in a formalism which gives an interesting characterization of computational processes, so-called CSP (*Communicating Sequential Process* [9]).

In CSP the word *process* stands for the behaviour pattern of objects. This concept comprehends sets of kinds of events or actions adequately chosen in order to give an description of such a behaviour pattern. The set of events associated to an object is called its *alphabet*.

Initially, we present intuitively descriptions of some concepts which we will use here. We follow the definitions given by Hoare [9]:

- The alphabet associated to an object, whose behaviour pattern is described by a process P, is denotated αP.

- $STOP_A$ is the process which never engages in any of the events of its alphabet A.

- $(x \rightarrow P)$ is the process which first engages in the event x and so behaves as described by P, provided that $x \in \alpha P$.

- A *trace* of the behaviour of a process is a finite sequence of symbols recording the events in which the process has engaged up to some moment in time.

- The most important operation on traces is *catenation*. The catenation of a trace s with a trace t, denoted by $s\hat{}t$, determines a trace by simply putting s and t together, in this order.

- A trace s is a *prefix* of a trace t ($s \leq t$), if s is a copy of an initial subsequence of t, i.e. there is an extension u of s such that the sequence constructed by the events of s followed by the events of u is the same as t . So, $\langle\rangle \leq t$ for any trace t, where $\langle\rangle$ is the empty trace.

- The sequential composition of the processes P and Q $(P;Q)$ is the process which first behaves like P; but when P terminates sucessfully, it continues by behaving like Q.

- The *concurrency* operator $\|$ concerns on the characterization of the interactions occurring between processes when they run concurrently. So, when it is applied to processes which have the same alphabet, it determines they interact with each other in a kind of a lock-step synchronization. When the operator $\|$ is applied to processes P and Q which have different alphabets, the events that are in both their alphabets require simultaneous participation of both P and Q; the events that are in αP but are not in αQ are no concern to Q, thus, P may engage in them independently of Q. Similarly, Q may engage alone in events which are in αQ but are not in αP.

- On the other hand, the *interleaving* operator $|||$ concerns on the characterization of the concurrently combination of processes with the same alphabet without directly interacting or synchronizing with each other. So, each event of a trace of the interleaving of two processes is an event which one of them can engage in it, however, if both processes can engage in a same event in a certain moment of time, the choice of which of them engage in such an event is nondeterministic.

  Each trace s of the interleaving of two processes P and Q can be constructed from a trace s_1 of P and a trace s_2 of Q by mixing their events in a nondeterministic way. So, we say that s *interleaves* s_1 and s_2.

- We must emphasize that processes with differents alphabets can be transformed in order to have the same alphabet without change the behaviour patterns that they describe. For instance, αP can be extended by including symbols which are only in αQ and we can obtain the process $P_{+(\alpha Q - \alpha P)} = (P \parallel STOP_{(\alpha Q - \alpha P)})$ which has exactly the same traces of P, but $\alpha P_{+(\alpha Q - \alpha P)} = \alpha P \cup \alpha Q$.

Lets denote $P_{+(\alpha Q-\alpha P)}$ as P_Q and $Q_{+(\alpha P-\alpha Q)}$ as Q_P

The denotation of the formulas in general will be given as processes in CSP. In order to give an interpretation to the linear tensor product and the operator $\odot$, we must take into account that the first one is usually considered as representing the absence of communication and is more restrict than the second one.

The concept of communication of processes is represented in CSP by means of the *concurrency* operator $\|$, but two processes might communicate only if their alphabets have a common event. So, we could obtain a denotation for the tensor product by means of the concurrency operator applied to processes which have disjoint alphabets. Thus, we must transform the original processes into processes with different alphabets. This can be done in CSP labelling each process by a different name, such that each event of a labelled process is also labelled by that name. In order to characterize perfectly the tensor product, which drives two premisses into only one, we make the labelled processes run concurrently until one of then stops.

In CSP, a process P labelled by l is denoted by $l : P$ and an event of the alphabet of $l : P$ is denoted by pairs $l.a$, where a is an event of the alphabet of P. So, we define the following operator in order to give a denotation for the tensor product.

Definition 4.1 *Let P and Q be processes, l_1 and l_2 distinct labels and d a non-labelled event, such that $d \notin \alpha P \cup \alpha Q$. Then, the operator $\bigwedge$ defined by*
$P \bigwedge Q = (l_1 : P\, ;\, d \rightarrow STOP \;\|\; l_2 : Q\, ;\, d \rightarrow STOP)$

Now, lets recall that the tensor product is more restrict than the operator $\odot$, then, it seems that the denotation for the operator $\odot$ could be given by means of the interleaving operator. Both of them are commutative and associative, moreover the interleaving operator captures very well the unrestricted combination of elements determined by the categorical interpretation of $\odot$.

On the other hand, the cartesian product & is usually considered as representing the intersection of elements. It will be denoted here by the interaction of processes by means of the concurrency operator, regarding that we must transform the original processes into processes with the same alphabet.

We will compare the denotations of the formulas by means of a kind of relation of inclusion defined in terms of traces of processes. However, such a relation will be little more complex than would be expected, since the traces of the denotation of the tensor product are particularly differents of the others: they have labelled events. So, we may have to deal with a trace s containing some events labelled by different labels $l_1, l_2, \cdots l_n$. It will not be a problem here, since we can eliminate those labels obtaining a *labels-free sequence* denoted $s \downarrow \{l_1, l_2, \cdots l_n\}$, like it is recursively defined below:

Definition 4.2 *Let s be a trace and L the set of labels which occur in s.*

(i) $\langle\rangle \downarrow L = \langle\rangle$

(ii) $(\langle l.a\rangle\hat{\ } s') \downarrow L = (\langle a\rangle\hat{\ } s') \downarrow L$, *if $s = \langle l.a\rangle\hat{\ } s'$.*

(iii) $(\langle a\rangle\hat{\ } s') \downarrow L = \langle a\rangle\hat{\ }(s' \downarrow L)$, *if a is a non-labelled event and $s = \langle a\rangle\hat{\ } s'$.*

Now, we are able to present more strictly the interpretation discussed above.

Definition 4.3 *Let* LLMS_c^* *be a fragment of* LLMS_c *which contains only the connectives* $\otimes$*,* $\odot$ *and* $\&$*.*
Let $[\![\ ]\!]$ *be a mapping, such that:*

- $[\![A]\!]$ *is a process,* *if* A *is atomic;*

- $[\![A \otimes B]\!] = [\![A]\!] \bigwedge [\![B]\!]$

- $[\![A \& B]\!] = [\![A]\!]_{[\![B]\!]} \parallel [\![B]\!]_{[\![A]\!]}$

- $[\![A \odot B]\!] = [\![A]\!]_{[\![B]\!]} ||| [\![B]\!]_{[\![A]\!]}$

- $[\![\Gamma]\!] = [\![\gamma]\!]$,
  where the formula γ *is obtained interpreting each occurrence of the operators "," and ";" in the context* Γ *as* $\otimes$ *and* $\odot$*, respectively;*

- $\Gamma \vdash C$ *is interpreted as* $[\![\Gamma]\!] \preceq [\![C]\!]$
  where the relation $P \preceq Q$*, reads* P **is subsumed by** Q*, is defined by:*
  $P \preceq Q$ *iff for each trace* s_1 *of* P*, there is a trace* s_2 *of* Q*, such that* $s_1 \downarrow L_1 \ \leq\ s_2 \downarrow L_2$*,*
  where L_1 *(*L_2*) is the set of labels which occur in* s_1 *(*s_2*, respectively).*

We have, hence, the following.

Theorem 4.1 (Soundness) *The mapping* $[\![\ ]\!]$ *preserves the sequent rules of* LLMS_c^**.*

Proof: For the sake of simplicity, when we will be dealing with the relation $\preceq$, instead of considering the original traces of the related denotations, we will consider directly the labels-free sequences obtained from those as representing such traces. It does not entail any lack of generality, since we can recover the original traces in a reverse way that we follow in order to elimiminate its occurrences of labels.

(i) $\otimes\mathcal{R}$

Suppose that $[\![\Gamma]\!] \preceq [\![A]\!]$ and $[\![\Delta]\!] \preceq [\![B]\!]$

Lets consider that $[\![\Gamma, \Delta]\!] = [\![\gamma \otimes \delta]\!]$ and s_1 is a trace of $[\![\gamma \otimes \delta]\!]$.

We must consider two cases:

(a) The trace s_1 is obtained by a successful termination of the process $[\![\gamma]\!]$, such that it includes some events which constitute a prefix of a trace of $[\![\delta]\!]$.
So, let $s_{1[\![\gamma]\!]}$ be a trace of determined by such a successful termination of $[\![\gamma]\!]$ and let $s_{1[\![\delta]\!]}$ be such a trace of $[\![\delta]\!]$.
Thus, by hypothesis, there is a trace s_2' of $[\![A]\!]$ such that $s_{1[\![\gamma]\!]} \leq s_2'$ and there is a trace s_2'' of $[\![B]\!]$ such that $s_{1[\![\delta]\!]} \leq s_2''$.
So, let s_2 be a sequence of events of $\alpha([\![A \otimes B]\!])$ which can be obtained from s_2' and s_2'' in such a way that:

- $s_1 \leq s_2$;
- s_2 contains all events of s_2', such that it ends with the last one; and
- s_2 includes only that events of s_2'' which also are in s_1.

Then, s_2 is a trace of $[\![A \otimes B]\!]$ obtained by a successful termination of $[\![A]\!]$ whose trace is s_2' , such that it includes some events which constitute a prefix of the trace s_2'' of $[\![B]\!]$. Therefore, $[\![\Gamma, \Delta]\!] \preceq [\![A \otimes B]\!]$.

(b) s_1 is obtained by a successful termination of $[\![\delta]\!]$, such that it includes some events which constitute a prefix of a trace of $[\![\gamma]\!]$.

This case is analogous to the previous one.

(ii) $\odot\mathcal{R}$

Suppose that $[\![\Gamma]\!] \preceq [\![A]\!]$ and $[\![\Delta]\!] \preceq [\![B]\!]$

Lets consider that $[\![\Gamma; \Delta]\!] = [\![\gamma \odot \delta]\!]$ and s_1 is a trace of $[\![\gamma \odot \delta]\!]$.

s_1 interleaves a trace of $[\![\gamma]\!]$ and a trace of $[\![\delta]\!]$. So, let $s_{1[\![\gamma]\!]}$ be such a trace of $[\![\gamma]\!]$ and $s_{1[\![\delta]\!]}$ such a trace of $[\![\delta]\!]$.

Thus, by hypothesis, there is a trace s_2' of $[\![A]\!]$ such that $s_{1[\![\gamma]\!]} \leq s_2'$ and there is a trace s_2'' of $[\![B]\!]$ such that $s_{1[\![\delta]\!]} \leq s_2''$.

Then, let s_2 be a sequence of events of $\alpha[\![A \odot B]\!]$ which interleaves s_2' and s_2'', in a such way that $s_1 \leq s_2$.

So, there is a trace s_2 of $[\![A \odot B]\!]$, such that $s_1 \leq s_2$.

Therefore, $[\![\Gamma; \Delta]\!] \preceq [\![A \odot B]\!]$.

(iii) $\odot\mathcal{L}$ and $\otimes\mathcal{L}$

These cases are directly proved from the semantic definition, since $[\![\Gamma[A, B]\,]\!] = [\![\Gamma[A \otimes B]\,]\!]$ and $[\![\Gamma[A; B]\,]\!] = [\![\Gamma[A \odot B]\,]\!]$.

(iv) $\&\mathcal{R}$

Suppose that $[\![\Gamma]\!] \preceq [\![A]\!]$ and $[\![\Gamma]\!] \preceq [\![B]\!]$

Let s_1 be a trace of $[\![\Gamma]\!]$. Thus, by hypothesis,

- there is a trace s_2' of $[\![A]\!]$ such that $s_1 \leq s_2'$; and
- there is a trace s_2'' of $[\![B]\!]$ such that $s_1 \leq s_2''$.

So, as $[\![A\&B]\!] = [\![A]\!]_{[\![B]\!]} \;||\; [\![B]\!]_{[\![A]\!]}$, there is a trace s_2 of $[\![A\&B]\!]$, such that $s_1 \leq s_2$.

Therefore, $[\![\Gamma]\!] \preceq [\![A\&B]\!]$.

(v) $1\&\mathcal{L}$ and $2\&\mathcal{L}$

The proofs of these cases are based on the facts that every trace of $[\![A\&B]\!]$ is a prefix of a trace of $[\![A]\!]$ and a trace of $[\![B]\!]$.

(vi) Structural Rule The main inferences enclosed by the structural are:

1.

$$\frac{\Delta[A;B] \vdash C}{\Delta[A,B] \vdash C}$$

The proof of this case follows of the proof that
$[\![A,B]\!] = [\![A \otimes B]\!] \preceq [\![A \odot B]\!] = [\![A;B]\!]$.

So, lets consider a trace s_1 of $[\![A \otimes B]\!]$.

We must consider two cases:

(a) The trace s_1 is obtained by a successful termination of the process $[\![A]\!]$, such that it includes some events which constitute a prefix of a trace of $[\![B]\!]$.

So, let $s_{1[\![A]\!]}$ be a trace of $[\![A]\!]$ determined by such a successful termination of $[\![A]\!]$ and let $s_{1[\![B]\!]}$ be such a trace of $[\![B]\!]$.

So, let s_2 be a sequence of events which can be obtained from $s_{1[\![A]\!]}$ and $s_{1[\![B]\!]}$ in a such way that s_2 interleaves $s_{1[\![A]\!]}$ and $s_{1[\![B]\!]}$, and $s_1 \leq s_2$.

Then, there is a trace s_2 of $[\![A \odot B]\!]$ such that $s_1 \leq s_2$. Therefore, $[\![A \otimes B]\!] \preceq [\![A \odot B]\!]$.

(b) s_1 is obtained by a successful termination of $[\![B]\!]$, such that it includes some events which constitute a prefix of a trace of $[\![A]\!]$.

This case is analogous to the previous one.

2.

$$\frac{\Delta[B,A] \vdash C}{\Delta[A,B] \vdash C} \qquad \frac{\Delta[B;A] \vdash C}{\Delta[A;B] \vdash C}$$

The proofs of these cases are based on the facts that the operators $\|$ and $|||$ are commutative.

3.

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

The proofs of these cases are based on the facts that the operators $\|$ and $|||$ are associative.

4.

$$(\mathbf{4.1})\frac{\Delta[(A,B);C] \vdash D}{\Delta[A,(B;C)] \vdash D} \qquad (\mathbf{4.2})\frac{\Delta[A;(B,C)] \vdash D}{\Delta[(A;B),C] \vdash D}$$

4.1 It is enough to prove that
$[\![A,(B;C)]\!] = [\![A \otimes (B \odot C)]\!] \preceq [\![(A \otimes B) \odot C]\!] = [\![(A,B);C]\!]$.

So, lets consider a trace s_1 of $[\![A \otimes (B \odot C)]\!]$

We must consider two cases:

(a) The trace s_1 is obtained by a successful termination of the process $[\![A]\!]$, such that it includes some events which constitute a prefix of a trace of $[\![B \odot C]\!]$.

So, let $s_{1[\![A]\!]}$ be a trace determined by such a successful termination of $[\![A]\!]$ and let $s_{1[\![B\odot C]\!]}$ be such a trace of $[\![B \odot C]\!]$.

On the other hand, $s_{1[\![B\odot C]\!]}$ interleaves a trace of $[\![B]\!]$ and a trace of $[\![C]\!]$.

So, let $s_{1[\![B]\!]}$ such a trace of $[\![B]\!]$ and $s_{1[\![C]\!]}$ such a trace of $[\![C]\!]$.

Then, let s_2 be a sequence which interleaves a trace s'_2 of $[\![A \otimes B]\!]$ and $s_{1[\![C]\!]}$, such that $s_1 \leq s_2$ and s'_2 which can be obtained from $s_{1[\![A]\!]}$ and $s_{1[\![B]\!]}$ in such a way that:

- s'_2 contains all events of $s_{1[\![A]\!]}$, such that it ends with the last one; and
- s'_2 includes only that events of $s_{1[\![B]\!]}$ which also are in s_1.

Then, there is a trace s_2 of $[\![(A \otimes B) \odot C]\!]$ such that $s_1 \leq s_2$.

Therefore, $[\![A \otimes (B \odot C)]\!] \preceq [\![(A \otimes B) \odot C]\!]$.

(b) s_1 is obtained by a successful termination of $[\![B \odot C]\!]$, such that it includes some events which constitute a prefix of a trace of $[\![A]\!]$.

This case is analogous to the previous one.

4.2 Analogous to the previous case.

5.

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

It is enough to prove that

$[\![(A; B), (C; D)]\!] = [\![(A \odot B) \otimes (C \odot D)]\!] \preceq [\![(A \otimes C) \odot (B \otimes D)]\!] = [\![(A, C); (B, D)]\!]$.

So, lets consider a trace s_1 of $[\![(A \odot B) \otimes (C \odot D)]\!]$

We must consider two cases:

(a) The trace s_1 is obtained by a successful termination of the process $[\![A \odot B]\!]$, such that it includes some events which constitute a prefix of a trace of $[\![C \odot D]\!]$.

So, let $s_{1[\![A\odot B]\!]}$ be a trace of $[\![A \odot B]\!]$ determined by such a successful termination of $[\![A \odot B]\!]$ and let $s_{1[\![C\odot D]\!]}$ be such a trace of $[\![C \odot D]\!]$.

Then, $s_{1[\![A\odot B]\!]}$ interleaves a trace of $[\![A]\!]$ and a trace of $[\![B]\!]$ and $s_{1[\![C\odot D]\!]}$ interleaves a trace of $[\![C]\!]$ and a trace of $[\![D]\!]$.

So, let $s_{1[\![A]\!]}$ such a trace of $[\![A]\!]$ and $s_{1[\![B]\!]}$ such a trace of $[\![B]\!]$; let $s_{1[\![C]\!]}$ such a trace of $[\![C]\!]$ and $s_{1[\![D]\!]}$ such a trace of $[\![D]\!]$.

Then, let s_2 be a sequence which interleaves a trace s'_2 of $[\![A \otimes C]\!]$ and a trace s''_2 of $[\![B \otimes D]\!]$ such that $s_1 \leq s_2$, where

– s'_2 which can be obtained from $s_{1[\![A]\!]}$ and $s_{1[\![C]\!]}$ in such a way that:

- s'_2 contains all events of $s_{1[\![A]\!]}$, such that it ends with the last one; and
- s'_2 includes only that events of $s_{1[\![C]\!]}$ which also are in s_1;

– s''_2 which can be obtained from $s_{1[\![B]\!]}$ and $s_{1[\![D]\!]}$ in such a way that:

- s''_2 contains all events of $s_{1[\![B]\!]}$, such that it ends with the last one; and

- s_2'' includes only that events of $s_{1[\![D]\!]}$ which also are in s_1.

Then, there is a trace s_2 of $[\![(A \otimes C) \odot (B \otimes D)]\!]$ such that $s_1 \leq s_2$.
Therefore, $[\![(A \odot B) \otimes (C \odot D)]\!] \preceq [\![(A \otimes C) \odot (B \otimes D)]\!]$

(b) s_1 is obtained by a successful termination of $[\![C \odot D]\!]$, such that it includes some events which constitute a prefix of a trace of $[\![A \odot B]\!]$.
This case is analogous to the previous one.

5 Conclusions

The motivation for this work came from O'Hearn's and Reddy's use of Linear Logic to deal with 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 de Paiva's dialectica categorical models GC 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 the 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 LLMS_c.

Furthermore, we gave a computational characterization of the operator $\odot$ by means of the *interleave* operator of CSP, while the tensor product $\otimes$ and the connective & were interpreted as restricted assynchronous concurrency and the interaction of processes, respectively.

But much work remains to be done. Firstly, going back to our original motivation, more work is needed to relate the system LLMS_c to O'Hearn's and Reddy's recent work[12, 14]. Secondly, from a logical pespective, we might want to use this system with two modalities, as a blueprint for work on relating modalities. Thirdly the interpretation of a fragment of LLMS_c in terms of CSP operators is useful, as it gives a very tangible explanation of our plethora of tensor products. But it would be more convincing if we could also explain the modalities in CSP terms. The domain-theoretical interpretation of the addressed fragment of LLMS_c can be built by using the standard event structure based semantics for CSP. Extending this to the whole of LLMS_c is also a topic for future work.

References

[1] J. Beck. Distributive laws. In *Seminar on Triples and Categorical Homoly Theory, LNM 80*, pages 119–140. Springer-Verlag, 1969.

[2] 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.

[3] 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).

[4] 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.

[5] 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.

[6] V. C. V. de Paiva. A category of multirelations. Technical Report 255, University of Cambridge Computer Laboratory, April 1991.

[7] V. C. V. de Paiva and J. M. E. Hyland. Generalising dialectica categories. (Revised version of a talk in the Montreal Category Theory Meeting 1991), 1995.

[8] J.-Y. Girard. Towards a geometry of interaction. In *Categories in Computer Science and Logic*, pages 69–108, Boulder, Colorado, June 1987. American Mathematical Society. (Contemporary Mathematics, Vol. 92).

[9] C. A. R. Hoare. *Communicating Sequential Processes.* Prentice-Hall International, London, 1985.

[10] Y. Lafont. The linear abstract machine. *Theoretical Computer Science*, 59:157–180, 1988.

[11] 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.

[12] 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.

[13] U. S. Reddy. A linear logic model of state. Eletronic manuscript (ftp),csc.uiuc.edu, directory /pub/reddy/papers, October 1993.

[14] 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.

[15] P. Wadler. Linear types can change the world! In *Programming Concepts and Methods*, Amsterdam, 1990. North-Holland. (Proc. IFIP TC 2 Working Conf, Sea of Galilee, Israel).

[16] P. Wadler. Is there a use for linear logic? In *Proc. Symp. on Partial Evaluation and Semantics-Based Program Manipulation, ACM*, pages 255–273, 1991. (SIGPLAN Notices, Sep. 1991).