

Flag: a Self-Dual Modality for Non-Commutative Contraction and Duplication in the Category of Coherence Spaces

Christian Retoré

LIRMM, Univ Montpellier & CNRS

*This paper is dedicated to the memory of my parents,
both math teachers, both deceased in 2021.*

After reminding what coherence spaces are and how they interpret linear logic, we define a modality “flag” in the category of coherence spaces (or hypercoherences) with two inverse linear (iso)morphisms: “duplication” from $(\text{flag } A)$ to $((\text{flag } A) \triangleleft (\text{flag } A))$ and “contraction” in the opposite direction — where \triangleleft is the self dual and non-commutative connective known as “before” in pomset logic and known as “seq(ential)” in the deep inference system (S)BV. In addition, as expected, the coherence space A is a retract of its modal image $(\text{flag } A)$. This suggests an intuitive interpretation of $(\text{flag } A)$ as “repeatedly A ” or as “ A at any instant” when “before” is given a temporal interpretation. We hope the semantic construction of $\text{flag}(A)$ will help to design proof rules for “flag” and we briefly discuss this at the end of the paper.

1 Presentation

Given the aftermath [10, 27, 15, 26] of pomset logic [18, 23] and of the related BV calculus of structures [8, 11, 12] we decided to publish this ancient work of ours [20], which answers a question of Jean-Yves Girard in [6]. This semantic work might help to find the proper rules for a self dual modality as recently sketched by Alessio Guglielmi in [10].

The structural rules of classical logic are responsible for the non-determinism of classical logic, and linear logic which carefully handles these rules is especially adequate for a constructive treatment of classical logic, as [6] shows. Linear logic handles structural rules by the modalities (a.k.a. exponentials) “?” and “!”. The modality “?” allows contraction and weakening in positive position, and the modality “!” in negative position. The formula $!A$ linearly implies A while the formula $?A$ is linearly implied by A . In semantical words, this means we have the linear morphisms:

$$\begin{array}{ccc}
 !A & \multimap & (!A \otimes !A) \\
 \downarrow \searrow & & \\
 1 & & A
 \end{array}
 \qquad
 \begin{array}{ccc}
 ?A & \multimap & ?A \wp ?A \\
 \downarrow \searrow & & \\
 1 & & A
 \end{array}$$

The major difficulty when dealing with classical logic are the cross-cuts, appearing in the cut elimination theorem of Gentzen as a rule called MIX [3]. This rule is a generalised cut between several occurrences of A and several occurrences of $\neg A$, i.e. a cut between two formulas both coming from contractions. This is a major cause of non-determinism: an example can be found in [6, Appendix B, Example 2, p. 294]. In linear logic, such a cut may not happen, since contraction only applies to $?A$ formulas, while their negation is $!A^\perp$ which can not come from a contraction — hence no such CUT is possible in linear logic.

Let us now quote the precise paragraph of [6, p. 257] which initially motivated our work on a self-dual modality for contraction:

The obvious candidate for a classical semantics was of course coherence spaces which had already given birth to linear logic; the main reason for choosing them was the presence of the involutive linear negation. However the difficulty with classical logic is to accommodate structural rules (weakening and contraction); in linear logic, this is possible by considering coherent spaces $?X$. But since classical logic allows contraction and weakening both on a formula and its negation, the solution seemed to require the linear negation of $?X$ to be of the form $?Y$, which is a nonsense (the negation of $?X$ is $!X^\perp$ which is by no means of isomorphic to a space $?Y$). Attempts to find a self-dual variant $\$Y$ of $?Y$ (enjoying $(\$Y)^\perp = \Y^\perp) systematically failed. The semantical study of classical logic stumbled on this problem of self-duality for years.

Here as well we focus on coherence spaces because of their tight relation to linear logic [5, 28, 19, 24, 7]. Once the modality is found in the category of coherence spaces, and shown to enjoy the expected properties, we briefly show that this modality also exists in the category of hypercoherences of Thomas Ehrhard [2], and discuss the possibility to conceive syntactic rules for this modality.

In our previous work on pomset logic [18, 23, 26], we studied a self-dual and non-commutative connective called “before”, written \triangleleft together with partially ordered multisets of formulae. This lead us to the modality \uparrow (flag) to be described in this paper. Flag is an endofunctor of the category of coherence spaces equipped with linear maps, flag is self-dual, and flag enjoys both *left and right contraction* with respect to “before” as a pair of linear isomorphisms $\uparrow A \circ \circ (\uparrow A \triangleleft \uparrow A)$ and A is a retract of $\uparrow A$. Fortunately, there is no weakening, which would yield unwanted morphisms like $1 \multimap A$ and $A \multimap 1$ for all A .

So far, there is no syntax extending pomset logic to this modality, although Alessio Guglielmi sketched a possible syntax in the closely related calculus of structures in [10]. In order to define a syntax, we should first study the basic steps of cut-elimination, in particular the contraction/contraction case and the commutative diagrams it requires, as Myriam Quatrini did in [17] for the logical calculus LC of [6]. We briefly discuss the guidelines for design of the rules for flag at the end of the paper.

2 Denotational Semantics of Linear Logic with “Before” Connective

This section is taken over from [26] and slightly adapted to the presentation of the flag modality. It is just a short presentation of coherences spaces and of the interpretation of linear logic in coherence spaces in order to provide a reasonably self-contained paper.

Definition 1 (categorical interpretation). *Denotational semantics or categorical interpretation of a logic is the interpretation in a category, here called \mathbf{C} :*

- *formulas are interpreted as objects in \mathbf{C}*
 - *an object $\llbracket a \rrbracket$ is arbitrarily associated with any atomic formula a (or propositional variable)*
 - *an n -ary logical connective (usually $n = 2$ or $n = 1$) $T[A_1, \dots, A_n]$ is interpreted as the construction of $\llbracket T[A_1, \dots, A_n] \rrbracket$ from the n objects $\llbracket A_1 \rrbracket \dots \llbracket A_n \rrbracket$.*
- *proofs are interpreted as morphisms of \mathbf{C} that are preserved under normalisation:*
 - *a proof d of $A \vdash B$ is interpreted as a morphism $\llbracket d \rrbracket$ from the object $\llbracket A \rrbracket$ to the object $\llbracket B \rrbracket$. There might as well be morphisms from $\llbracket A \rrbracket$ to $\llbracket B \rrbracket$ that are not the interpretation of a proof of $A \vdash B$ unless the interpretation is proved to be “fully abstract” (some also say “total”).*

- whenever a proof d of $A \vdash B$ reduces to a proof d' (hence a proof of $A \vdash B$ as well) by (the transitive closure) of β -reduction or cut-elimination one has $\llbracket d \rrbracket = \llbracket d' \rrbracket$ (hence the name denotational semantics).
- when there is a terminal object $\mathbb{1}$, a semantic entity in the object (formula) X of \mathbf{C} can be viewed as a morphism from $\mathbb{1}$ to X . Semantic entities include the interpretations of the proofs of $\vdash X$ but there might be other semantic entities in X , that are not the interpretation of any proof of $\vdash X$ when the interpretation is not fully abstract.

Categorical interpretations of intuitionistic logic take places in Cartesian closed categories while categorical interpretations of linear logic take place in a monoidal closed category — with monads for interpreting the modalities (also called exponentials) of full linear logic.

2.1 Coherence Spaces

The category of coherence spaces is a concrete category: objects are (countable) sets endowed with a binary relation, and morphisms are linear maps. It interprets the proofs up to cut-elimination (or β reduction). Coherence spaces are tightly related to linear logic: indeed, linear logic arose from this particular semantics, initially invented to model second order lambda calculus i.e. quantified propositional intuitionistic logic [4]. Coherence spaces are themselves inspired from the categorical work on ordinals by Jean-Yves Girard; they are the binary qualitative domains.

Definition 2 (coherence space). *A coherence space A is a set $|A|$ (possibly infinite) called the web of A whose elements are called tokens, endowed with a binary reflexive and symmetric relation called coherence on $|A| \times |A|$ written $\alpha \supset \alpha'[A]$ or simply $\alpha \supset \alpha'$ when A is clear.*

The following notations are common and useful:

$$\begin{aligned} \alpha \frown \alpha'[A] &\text{ iff } \alpha \supset \alpha'[A] \text{ and } \alpha \neq \alpha' \\ \alpha \asymp \alpha'[A] &\text{ iff } \alpha \not\supset \alpha'[A] \text{ or } \alpha = \alpha' \\ \alpha \smile \alpha'[A] &\text{ iff } \alpha \not\supset \alpha'[A] \text{ and } \alpha \neq \alpha' \end{aligned}$$

A proof of A is to be interpreted by a *clique* of the corresponding coherence spaces A , a *clique* being a set of pairwise coherent tokens in $|A|$ — we write $x \in A$ for $x \subset |A|$ and for all α, α' in x one has $\alpha \supset \alpha'$. Observe that for all $x \in A$, if $x' \subset x$ then $x' \in A$.

Definition 3 (linear morphism). *A linear morphism F from A to B is a morphism mapping cliques of A to cliques of B such that:*

- For all $x \in A$ if $(x' \subset x)$ then $F(x') \subset F(x)$
- For every family $(x_i)_{i \in I}$ of pairwise compatible cliques — that is to say $(x_i \cup x_j) \in A$ holds for all $i, j \in I$ — $F(\cup_{i \in I} x_i) = \cup_{i \in I} F(x_i)$.¹
- For all $x, x' \in A$ if $(x \cup x') \in A$ then $F(x \cap x') = F(x) \cap F(x')$ – this last condition is called *stability*.

Linear morphisms are maps in the set theoretic sense and they compose as maps.

Due to the removal of structural rules, linear logic has two kinds of conjunction:

$$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} \otimes \quad \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B} \&$$

¹The morphism is said to be *stable* when this second condition is replaced with $F(\cup_{i \in I} x_i) = \cup_{i \in I} F(x_i)$ holds more generally for the union of a directed family of cliques of A , i.e. $\forall i, j \exists k (x_i \cup x_j) = x_k$.

Those two rules are equivalent when contraction and weakening are allowed. Multiplicative conjunction splits contexts as in the \otimes rule above while the additive conjunction duplicates context as in the $\&$ rule above. Regarding denotational semantics, the web of the coherence space associated with a formula $A * B$ with $*$ a binary *multiplicative* connective is the Cartesian *product* of the webs of A and B i.e. $|A * B| = |A| \times |B|$ — the web of a binary *additive* connective $A \square B$ is the *disjoint union* (also called *sum*) of the webs of A and B i.e. $|A \square B| = |A| \uplus |B|$.

Definition 4. *Negation is a unary connective which is both multiplicative and additive:*²

$$|A^\perp| = |A| \text{ and } \alpha \circledast' \alpha[A^\perp] \text{ iff } \alpha \asymp \alpha'[A]$$

One may wonder how many binary multiplicative connectives there are, i.e. how many different coherence relations one may define on $|A| \times |B|$ from the coherence relations on A and on B .

We can limit ourselves to the connectives $*$ that are covariant functors in both A and B , i.e. the ones such that

- $\alpha \circledast \alpha'[A]$ and $\beta \circledast \beta'[B]$ entails $(\alpha, \beta) \circledast (\alpha', \beta')[A * B]$
- $\alpha \asymp \alpha'[A]$ and $\beta \asymp \beta'[B]$ entails $(\alpha, \beta) \asymp (\alpha', \beta')[A * B]$

Indeed, there is a negation, hence a contravariant connective in A is a covariant connective in A^\perp . Hence when both components are in the \asymp relationship so are the two couples, and when they are both coherent, so are the two couples. If a connective $A * B$ is not like this then $A^\perp * B$ or $A * B^\perp$ or $A^\perp * B^\perp$ is as we wish.

As it is easily observed, given two tokens α, α' in a coherence space C exactly one of the three following properties hold:

$$\alpha \smile \alpha' \quad \alpha = \alpha' \quad \alpha \frown \alpha'$$

In order to define a multiplicative connective, one should simply say when $(\alpha, \beta) \circledast (\alpha', \beta')[A * B]$ holds depending on whether $\alpha \circledast \alpha'[A]$ and $\beta \circledast \beta'[B]$ hold. Thus defining a binary multiplicative connective is to fill a nine cell table as the ones below, the first column indicates the relation between α and α' in A , while the first row indicates the relation between β and β' in B .

However if $*$ is assumed to be covariant in both its arguments, seven out of the nine cells are filled, so the only free values are the ones in the right upper cell, NE=North-East, and the one in the left bottom cell SW=South-West. They cannot be = so it makes four possibilities.

$A * B$		∪	=	∩
∪		∪	∪	NE?
=		∪	=	∩
∩		SW?	∩	∩

If one wants $*$ to be commutative, there are only two possibilities, namely $NE = SW = \frown$ (\wp) and $NE = SW = \smile$ (\otimes).

<table border="1" style="border-collapse: collapse; text-align: center; width: 100px;"> <tr> <td style="border: none;">$A \wp B$</td> <td style="border: none;"> </td> <td style="border: none;">∪</td> <td style="border: none;">=</td> <td style="border: none;">∩</td> </tr> <tr> <td style="border: none;">∪</td> <td style="border: none;"> </td> <td style="border: none;">∪</td> <td style="border: none;">∪</td> <td style="border: none;">∩</td> </tr> <tr> <td style="border: none;">=</td> <td style="border: none;"> </td> <td style="border: none;">∪</td> <td style="border: none;">=</td> <td style="border: none;">∩</td> </tr> <tr> <td style="border: none;">∩</td> <td style="border: none;"> </td> <td style="border: none;">∩</td> <td style="border: none;">∩</td> <td style="border: none;">∩</td> </tr> </table>	$A \wp B$		∪	=	∩	∪		∪	∪	∩	=		∪	=	∩	∩		∩	∩	∩	and	<table border="1" style="border-collapse: collapse; text-align: center; width: 100px;"> <tr> <td style="border: none;">$A \otimes B$</td> <td style="border: none;"> </td> <td style="border: none;">∪</td> <td style="border: none;">=</td> <td style="border: none;">∩</td> </tr> <tr> <td style="border: none;">∪</td> <td style="border: none;"> </td> <td style="border: none;">∪</td> <td style="border: none;">∪</td> <td style="border: none;">∪</td> </tr> <tr> <td style="border: none;">=</td> <td style="border: none;"> </td> <td style="border: none;">∪</td> <td style="border: none;">=</td> <td style="border: none;">∩</td> </tr> <tr> <td style="border: none;">∩</td> <td style="border: none;"> </td> <td style="border: none;">∪</td> <td style="border: none;">∩</td> <td style="border: none;">∩</td> </tr> </table>	$A \otimes B$		∪	=	∩	∪		∪	∪	∪	=		∪	=	∩	∩		∪	∩	∩
$A \wp B$		∪	=	∩																																						
∪		∪	∪	∩																																						
=		∪	=	∩																																						
∩		∩	∩	∩																																						
$A \otimes B$		∪	=	∩																																						
∪		∪	∪	∪																																						
=		∪	=	∩																																						
∩		∪	∩	∩																																						

However if we do not ask for the connective $*$ to be commutative we have a third *non-commutative* connective $A \triangleleft B$ and a fourth connective $A \triangleright B$ which is simply $B \triangleleft A$.

²Indeed, its web is both the Cartesian product of the web of A and the disjoint union of the web of A .

$A \triangleleft B$				
∪				
=		=		∩
∩		∩		∩

and

$A \triangleright B$				
∪		∪		∩
=		∪	=	∩
∩		∪		∩

Definition 5 (multiplicative connectives). *The coherence relation for the multiplicative connectives are defined as follows:*

- $(\alpha, \beta) \circ (\alpha', \beta')[A \otimes B]$ iff $\alpha \circ \alpha'[A]$ and $\beta \circ \beta'[B]$
- $(\alpha, \beta) \circ (\alpha', \beta')[A \wp B]$ iff $\alpha \circ \alpha'[A]$ or $\beta \circ \beta'[B]$
- $(\alpha, \beta) \circ (\alpha', \beta')[A \triangleleft B]$ iff $\alpha \cap \alpha'[A]$ or $(\alpha = \alpha'$ and $\beta \circ \beta'[B])$

Definition 6 (trace of a linear morphism). *The linear morphisms from A to B written $\text{Hom}(A, B)$ can be represented by the coherence space $A \multimap B = A^\perp \wp B$ ³*

$A \multimap B$				
∪		∩		∩
=		∪	=	∩
∩		∪		∩

Linear morphisms are in a one-to-one correspondence with cliques of $A \multimap B$. Given a clique $f \in (A \multimap B)$ the map F_f from cliques of A to cliques of B defined by $F_f(x) = \{\beta \in |B| \mid \exists \alpha \in x \text{ such that } (\alpha, \beta) \in f\}$ is a linear morphism. Conversely, given a linear morphism F , the set $\{(\alpha, \beta) \in |A| \times |B| \mid \beta \in F(\{\alpha\})\}$ called the trace of F is a clique of $A \multimap B$.

Proposition 7. *The connective \triangleleft is (1) non commutative, (2) associative, (3) self-dual and (4) lays in between \otimes and \wp .*

Proof. 1. \triangleleft is non commutative. From the definition of the coherence space $A \triangleleft B$ it is clear that there is no canonical isomorphism between $A \triangleleft B$ and $B \triangleleft A$, hence $A \triangleleft B$ and $B \triangleleft A$ are not isomorphic in general.

2. \triangleleft is associative. The set $\{((\alpha, (\beta, \gamma)), ((\alpha, \beta), \gamma)) \mid \alpha \in |A|, \beta \in |B|, \gamma \in |C|\}$ defines a linear isomorphism from $A \triangleleft (B \triangleleft C)$ to $(A \triangleleft B) \triangleleft C$.
3. \triangleleft is self-dual, $(A \triangleleft B)^\perp \equiv (A^\perp) \triangleleft (B^\perp)$. Given two different tokens (α, β) and (α', β') in $|A| \times |B|$, observe that:

- (a) $(\alpha, \beta) \cap (\alpha', \beta')[A \triangleleft B]$ means $(\alpha = \alpha'$ and $\beta \cap \beta'[B])$ or $\alpha \cap \alpha'[A]$
- (b) $(\alpha, \beta) \cap (\alpha', \beta')[A^\perp \triangleleft B^\perp]$ means $(\alpha = \alpha'$ and $\beta \cup \beta'[B])$ or $\alpha \cup \alpha'[A]$

Given that those two tokens (α, β) and (α', β') are different, either:

- If $\alpha \neq \alpha'$ then either
 - $\alpha \cap \alpha'[A]$, 1 holds and 2 does not hold or
 - $\alpha \cup \alpha'[A]$, 2 holds and 1 does not hold.
- If $\alpha = \alpha'$ then $\beta \neq \beta'$ and either
 - $\beta \cap \beta'[B]$, 1 holds and 2 does not hold or
 - $\beta \cup \beta'[B]$ 2 holds and 1 does not hold.

³This internalisation of $\text{Hom}(A, B)$ makes the category monoidal closed, but not Cartesian closed because the associated conjunction, namely \otimes is not a product.

Consequently, any two different tokens (α, β) and (α', β') in $|A| \times |B|$ are either strictly coherent in $(A \triangleleft B)$ or are strictly coherent in $A^\perp \triangleleft B^\perp$. Therefore $(A \triangleleft B)^\perp \equiv (A^\perp) \triangleleft (B^\perp)$.

4. $(A \otimes B) \multimap (A \triangleleft B) \multimap (A \wp B)$ The set $\{((\alpha, \beta), (\alpha, \beta)) \mid \alpha \in |A|, \beta \in |B|\}$ defines a linear morphism from $A \otimes B$ to $A \triangleleft B$ and the very same set of pairs of tokens also defines a linear morphism from $A \triangleleft B$ to $A \wp B$.

□

The definition of the coherence spaces associated with $A \triangleleft B$ and $A \wp B$ can be generalised to series parallel (SP) partial orders of formulas, a well-known class of finite partial orders defined as follows. Given two partial orders $O_1 \subset |O_1|^2$ and $O_2 \subset |O_2|^2$ on two disjoint sets $|O_1|$ and $|O_2|$, one can defined two partial orders on $|O_1| \uplus |O_2|$:

- $O_1 \wp O_2$ the parallel composition of O_1 and O_2 defined by $O_1 \wp O_2 \subset (|O_1| \uplus |O_2|)^2$ is the disjoint union of the two partial orders O_1 and O_2 viewed as two sets of couples
- $O_1 \triangleleft O_2$ the series composition of O_1 and O_2 defined by $O_1 \triangleleft O_2 \subset (|O_1| \times |O_2|) \subset (|O_1| \uplus |O_2|)^2$

The class of series-parallel partial orders is the the smallest class of binary relations defined from the one element orders and closed by two binary operations \wp and \triangleleft on partial orders defined above.

Definition 8 (sp-order of coherence spaces). *Given an SP-order O on $\{1, \dots, n\}$ and a multiset of formulas $\{A_1, \dots, A_n\}$ (one may have $A_i = A_j$ as formulas for $i \neq j$) one may consider the formula $O(A_1, \dots, A_n)$ defined from A_1, \dots, A_n using only the \wp and \triangleleft connectives, following the order O : series composition in O corresponds to the \triangleleft connective in $O(A_1, \dots, A_n)$ and parallel composition in O corresponds to the \wp connective in $O(A_1, \dots, A_n)$. One can define directly the coherence space corresponding to the formula $O(A_1, \dots, A_n)$ as follows:*

web: $|A_1| \times \dots \times |A_n|$

strict coherence: $(\alpha_1, \dots, \alpha_n) \frown (\alpha'_1, \dots, \alpha'_n) [O[A_1, \dots, A_n]]$

whenever there exists i such that $\alpha_i \frown \alpha'_i$ and for every $j <_O i$ one has $\alpha_j = \alpha'_j$.

Coming back to what a categorical interpretation is, let us define precisely the category where our categorical interpretation takes place:

Definition 9 (The category **Coh**). *The category of coherence spaces is defined by its objects which are coherence spaces and its linear morphisms which are linear maps. It contains constructions corresponding to negation and to the connectives $\wp, \otimes, \triangleleft$. Linear maps from A to B can be internalised as their trace, that are object of the coherence space $A \multimap B$.*

Linear logic is issued from coherence semantics, and consequently coherence semantics is close to linear logic syntax. Coherence spaces may even be turned into a fully abstract (a.k.a. total) model in the multiplicative case (without before), see [14]. The before connective and pomset logic are also issued from coherence semantics, so let us define precisely the categorical interpretation of pomset logic, the logic that the flag modality is extending.

2.2 A Sound and Faithful Interpretation of Pomset Proof-Nets in Coherence Spaces

We cannot present here pomset proof-nets in full detail, but they are thoroughly presented in the recent paper [26]. Nevertheless we can briefly present handsome proof-nets for pomset logic. This view of proof-nets was initially introduced in [22] for MLL, and in [25] for pomset logic.

De Morgan laws hold for pomset logic:

$$\begin{aligned}
(A^\perp)^\perp &= A \\
(A \wp B)^\perp &= (A^\perp \otimes B^\perp) \\
(A \triangleleft B)^\perp &= (A^\perp \triangleleft B^\perp) \\
(A \otimes B)^\perp &= (A^\perp \wp B^\perp)
\end{aligned}$$

Because of the De Morgan laws above, as in classical logic, every formula does have an equivalent negation normal form (in which negation only appears on atoms), and proof-net syntax only use negation normal forms. The set of pomset formulas in negation normal form F over a set of propositional variables P is defined as follows:

$$F := P \mid P^\perp \mid F \wp F \mid F \triangleleft F \mid F \otimes F$$

The *directed cograph* or *dicograph* for short $\text{dicog}(F)$ associated with a formula F is defined as follows. Given a formula F we say that two occurrences of atoms a and b (elements of $P \uplus P^\perp$) in F meet on the connective $*$ ($*$ being one of $\wp, \triangleleft, \otimes$) whenever F is $H[G[a] * G[b]]$ where $E[\]$ denotes a formula with a hole — beware that the order matters, a and b meet on $*$ does not mean that b and a meet on $*$, one connective being non commutative. The arcs of $\text{dicog}(F)$ are defined as follows:

- there is an R (for Red or Regular, when printed in black and white) arc from a to b and no R arc from b to a whenever a and b meet on a “before”,
- there is an R edge (that is a pair of opposite R arcs) between a and b whenever a and b meet on a tensor,
- there is no R arc from a to b nor from b to a when a and b meet on a par

Several distinct formulas may describe the same dicograph: in this case the formulas only differ up to the associativity and commutativity of \wp, \otimes and to the associativity of \triangleleft , e.g. $(A \otimes (B \triangleleft (C \triangleleft D))) \otimes E$ and $(E \otimes A) \otimes ((B \triangleleft C) \triangleleft D)$ define the same dicograph. Dicographs are directed graphs that are the dicograph of some formula. Cographs are dicographs with no directed edges, i.e. the dicograph corresponding to a formula without any \triangleleft connective. Series-parallel partial orders (already mentioned in subsection 2.1) are the dicographs with no undirected edges, corresponding to formulas with no \otimes connectives. For more details on those structures and their characterisation by absence of certain subgraphs see [1] or the more recent book chapters [26, 13]

A *pomset proof structure* with conclusion F is an edge bicoloured graph (Red or Regular arcs and Blue or Bold edges) over the occurrences of atoms of F consists in the R dicograph $\text{dicog}(F)$ and a set of B edges satisfying: no two B edges are adjacent, every atom (vertex) of F is incident to a (unique) B edge, and the end vertices of B edges are dual atoms a and a^\perp . Notice that the conclusion of a proof structure contains the same number of occurrences of a and of occurrences of a^\perp for each propositional variable a . The conclusion F of a proof structure π may be written in several ways as $G[C_1, \dots, C_n]$ with G only using the connectives \wp, \triangleleft : in this case one may also say that the conclusions of π are C_1, \dots, C_n ordered by the series parallel partial order described by G .

A *pomset proof-net* or a correct pomset proof structure is a proof structure when it satisfies the *correctness criterion*: every elementary circuit (directed cycle) alternating R and B edges (AE-circuit for short) contains an R chord (an R edge or arc not on the circuit but whose end vertices are on the circuit).

It is possible to interpret a proof-net and even a simple proof structure with conclusion T (a formula or a dicograph of atoms) as a set of tokens of the corresponding coherence space T . So the fact that the interpretation should be a clique in the coherence space $\llbracket T \rrbracket$ and also cut-elimination guide the design of the deductive systems for pomset logic [26].



Figure 1: Two incorrect handsome proof structures namely $((a \otimes c^\perp) \wp (a^\perp \otimes c)) \triangleleft (b \wp b^\perp)$ and $((a \triangleleft c^\perp) \wp (c \triangleleft a^\perp)) \triangleleft (b \wp b^\perp)$. Both are incorrect because of the chordless AE-circuit: $a, c^\perp, c, a^\perp, a$. Red or Regular arcs represent the conclusion formula (dicograph) while Blue or Bold edges represent the axioms.



Figure 2: Two correct handsome proof structures (or proof-nets) $((a \wp a^\perp) \otimes (c^\perp \wp c)) \triangleleft (b \wp b^\perp)$ and $((a \triangleleft c^\perp) \wp (a^\perp \triangleleft c)) \triangleleft (b \wp b^\perp)$ — none of them contains a chordless AE-circuit. Red or Regular arcs represent the conclusion formula (dicograph) while Blue or Bold edges represent the axioms.

Computing the semantics of a proof-net is rather easy, using Girard's experiments — we rather define experiments from axioms to conclusions as in [19, 24] but it makes no difference. We define the interpretation of a proof structure, which is not necessarily a proof-net as a set of tokens of the web of the conclusion formula. Assume the axioms of a proof structure is the set of B -edges $B = \{a_i - a_i^\perp \mid 1 \leq i \leq n\}$ and that each of the a_i has a corresponding coherence space also denoted by a_i . For each a_i choose a token $\alpha_i \in |a_i|$. If the conclusion of the proof structure is a dicograph T then replacing the two occurrences a_i and a_i^\perp that are connected with an axiom (linked with a B edge) with α_i yields a term, which when converting $x * y$ (with $*$ being one of the connectives, $\wp, \triangleleft, \otimes$) with (x, y) , yields a token in the web of the coherence space associated with T — this token in $|T|$ is called the *result* of the experiment.

Given a normal (cut-free) proof structure π with conclusion T the interpretation $\llbracket \pi \rrbracket$ of the normal proof structure π is the set of all the results of the experiments on π . As initially shown for proof-nets with links in [19] or for handsome proof nets in [26]:

Theorem 10. *A proof structure π with conclusion T is a proof-net (contains no chordless AE-circuit) if and only if its interpretation $\llbracket \pi \rrbracket$ is a clique of the coherence space T — remember that only cliques of T can be the interpretation of some proof of T .*

When π is not normal, i.e. includes cuts that are formulas $K \otimes K^\perp$, not all experiments succeed and provide results: an experiment is said to succeed when in every cut $K \otimes K^\perp$ (a cut may be viewed as a conclusion $K \otimes K^\perp$) the value α on an atom a of the cut K is the same as the value on the corresponding atom a^\perp in K^\perp . Otherwise the experiment fails and has no result. The set of the results of all succeeding

experiments of a proof *net* π is a clique of the coherence space T . It is the interpretation $\llbracket \pi \rrbracket$ of the normal proof-net π . We can prove, that whenever π reduces to π' by cut elimination $\llbracket \pi \rrbracket = \llbracket \pi' \rrbracket$, see e.g. [19].

3 The “Flag” Modality in Coherence Spaces

The construction of the coherence space $\lceil A \rceil$ is somehow inspired by the fact that the web $|!A|$ of $!A$ consists in the finite sets of elements in the web $|A|$. The intuition is that the web of $\lceil A \rceil$ consists in linearly ordered copies of the web of A , the underlying order being equivalent to two ordered copies of it, so that $\lceil A \rceil$ is isomorphic to $\lceil A \rceil \triangleleft \lceil A \rceil$, i.e. the web $|\lceil A \rceil|$ is $\dots \triangleleft |A| \triangleleft |A| \triangleleft |A| \triangleleft |A| \triangleleft |A| \triangleleft \dots$. For the order to be equivalent to two copies of it, we first succeeded with \mathbb{Q} copies of A , but elements of $|\lceil A \rceil|$ were lacking a finite description. Then, following a suggestion by Achim Jung, we looked at 2^ω copies of $|A|$ and even better at the continuous functions from 2^ω to $|A|$ endowed with the discrete topology on $|A|$ and that way elements of the web $|\lceil A \rceil|$ of $\lceil A \rceil$ have a finite description.

3.1 Remarks on the Continuous Functions from the Cantor Space to a Discrete Space

Let us write 2 for $\{0, 1\}$ and 2^* for the set of finite words on 2 , including the empty word, 2^ω for the set of infinite words on 2 . Letters like w, v, u range over 2^ω , while m range over 2^* . The expression $w = m(m')^\omega$ stands for $w = mm'm'm'm'm' \dots$

The set 2^ω of infinite words on 2 is assumed to be endowed with:

- the usual total lexicographical order on infinite strings defined by:

$$w_1 < w_2 \text{ iff } \exists m \in 2^* \exists w'_1, w'_2 \in 2^\omega \ w_1 = m0w'_1 \text{ and } w_2 = m1w'_2$$

- the usual product topology generated by the basis of clopen sets $(U_m)_{m \in 2^*}$ with

$$U_m = \{w \in 2^\omega \mid \exists w' \in 2^\omega \ w = mw'\}$$

A well-known result on 2^ω is:

Proposition 11. *A clopen set U of 2^ω always is a finite union of base clopen sets, and thus a clopen set always has a minimum element (and a maximal element as well).*

Proof. As U is open, it is a union of base clopen set say $(O_i)_{i \in I}$. But as U is closed (clopen) in a compact, U is compact, and from the covering of U by the $(O_i)_{i \in I}$, there exists a finite covering $(O_{i_k})_{k \in K}$ of U by base clopen sets with K finite. Base clopen sets do have a minimum element, and the minimum of those finitely many minimums is the minimum of U . \square

Given a continuous function f from 2^ω to M endowed with the discrete topology, $f^{-1}\{a\}$ is a clopen set of 2^ω and $\cup_{a \in M} f^{-1}\{a\} = 2^\omega$.

Because 2^ω is compact a finite open cover can be extracted from $\cup_{a \in M} f^{-1}\{a\} = 2^\omega$ so there exists a finite number n such that $\cup_{1 \leq i \leq n} f^{-1}\{a_i\} = 2^\omega$, and replacing each clopen $f^{-1}\{a_i\}$ by a finite union of base clopen sets, we obtain a finite cover by base clopen sets U_k , $1 \leq k \leq p$, with f constant on each U_m .

From this one easily obtains:

Proposition 12. (gt_M generic trees on M) The set gt_M of continuous functions from 2^ω to a set M (discrete topology) is in a one-to-one correspondence with the set of finite binary trees on M such that any two sister leaves have distinct labels.

Thus, an element f of gt_M may either be described:

1. As a finite set $\{(m_1, a_1), \dots, (m_k, a_k)\} \subset \mathcal{P}_{\text{fin}}(2^* \times M)$ satisfying:

$$(a) \quad \forall w \in 2^\omega \exists ! i \leq k \exists w' \in 2^\omega \quad w = m_i w'$$

$$(b) \quad \forall i, j \leq k \left[\exists m \in 2^* \quad m_i = m0 \text{ and } m_j = m1 \right] \Rightarrow a_i \neq a_j$$

In this formalism $f(w)$ is computed as follows: applying (a), there exists a unique i such that $w = m_i w'$ for some w' , let $f(w) = f(m_i w') = f(m_i(0)^\omega) = f(m_i(1)^\omega) = a_i$

2. As the normal form of a term of the following grammar where \underline{M} stands for $\{\underline{x} \mid x \in M\}$ ⁴:

$$\mathcal{T}_M ::= \underline{M} \mid \langle \mathcal{T}_M \mathcal{T}_M \rangle$$

where the reduction is $\forall x \in M \quad t[\langle \underline{x} \underline{x} \rangle] \longrightarrow t[\underline{x}]$ where $t[u]$ with u in \mathcal{T}_M means a term of \mathcal{T}_M having an occurrence of the subterm $u \in \mathcal{T}_M$. In this formalism $f(w)$ is computed as follows:

$$\begin{aligned} f &= \underline{a} & : & \quad f(w) = a \\ f &= \langle t_0 t_1 \rangle & : & \quad \begin{aligned} f(0w) &= t_0(w) \\ f(1w) &= t_1(w) \end{aligned} \end{aligned}$$

Example: Let $M = \{a, b, c\}$ Here are the three description of the same element of gt_M :

0. As a function

$$\begin{aligned} f(000w) &= a & f(100w) &= a \\ f(001w) &= a & f(101w) &= b \\ f(010w) &= a & f(110w) &= a \\ f(011w) &= a & f(111w) &= b \end{aligned}$$

1. as a finite set of pairs $\{(m_i, a_i) \mid m_i \in 2^* \text{ and } a_i \in M\}$:

$$f = \{(0, a), (100, a), (101, b), (110, a), (111, b)\}$$

2. as a normal term of \mathcal{T}_M : $f = \langle \underline{a} \langle \langle \underline{a} \underline{b} \rangle \langle \underline{a} \underline{b} \rangle \rangle \rangle$ such a term is the normal form of, e.g. $\langle \langle \underline{a} \underline{a} \rangle \langle \langle \underline{a} \underline{b} \rangle \langle \langle \underline{a} \underline{a} \rangle \langle \underline{b} \underline{b} \rangle \rangle \rangle \rangle$

Here is one more easy remark:

Proposition 13. Let f, g be two functions in gt_M . If $f \neq g$ then there exists $w \in 2^\omega$ such that

$$f(w) \neq g(w) \text{ and } \forall w' < w \quad f(w') = g(w')$$

Proof. The product of the discrete topological space M by itself is the discrete topological space over $M \times M$. Hence the function Δ from $M \times M$ to 2 defined by $\Delta(x, y) = 1$ iff $x = y$ is continuous. The function (f, g) from 2^ω to $M \times M$ defined by $(f, g)(w) = (f(w), g(w))$ is continuous, because we use the product topology on $M \times M$. Therefore $(\Delta \circ (f, g))^{-1}(0)$ is a clopen set, which has a lowest element w ending with an infinite sequence of 0 (as observed in proposition 11), i.e. $w = m(0)^\omega$. Thus this w enjoys $f(w) \neq g(w)$ and $f(v) = g(v)$ whenever $v < w$. \square

⁴In order to distinguish between the constant function \underline{a} mapping every word of 2^ω onto $a \in M$ and $a \in M$.

3.2 The “Flag” Modality

As said in previous section, after a first solution with rational numbers in $[0, 1[$, following a suggestion by Achim Jung, we arrived to the following solution:

Definition 14. *Let A be a coherence space. We define $\uparrow A$ as follows:*

web $|\uparrow A| = \text{gt}_{|A|}$ *the set of continuous functions from $\mathcal{2}^\omega$ to $|A|$ (discrete topology).*

coherence *Two functions f and g of $\text{gt}_{|A|} = |\uparrow A|$ are said to be strictly coherent whenever*

$$\exists w \in \mathcal{2}^\omega f(w) \frown g(w)[A] \text{ and } \forall w' < w \ f(w') = g(w')$$

Observe that the above definition generalises to the infinite case $A_1 \triangleleft A_2 \triangleleft \dots \triangleleft A_n$ with $A_i = A$ for all i as defined in definition 8 of the coherence space associated with an SPordered collection of coherence space.

3.3 Properties

The following is clear from proposition 12:

Proposition 15. *(denumerable web) If $|A|$ is denumerable, so is $|\uparrow A|$.*

And next come a key property:

Proposition 16. *(self-duality) The modality \uparrow is self-dual, i.e. $(\uparrow A)^\perp \equiv \uparrow(A^\perp)$*

Proof. Those two coherence spaces obviously have the same web. Hence it is equivalent to show that, given two distinct tokens f, g in $|\uparrow A|$ either $f \frown g[\uparrow A]$ (*) holds or $f \frown g[\uparrow(A^\perp)]$ (**) holds. If $f \neq g$, then, because of previous proposition 13, there exists an infinite word w in $\mathcal{2}^\omega$ such that :

$$f(w) \neq g(w) \text{ and } \forall w' < w \ f(w') = g(w').$$

Therefore,

- either $[f(w) \frown g(w)[A] \text{ and } \forall w' < w \ f(w') = g(w')]$ holds
- or $[f(w) \frown g(w)[A^\perp] \text{ and } \forall w' < w \ f(w') = g(w')]$ holds.

This is component-wise equivalent to the expected exclusive disjunction (*) or (**). □

Proposition 17. *(contraction isomorphism) There is a canonical linear isomorphism $\uparrow A \circ \dashv \dashv \uparrow A \triangleleft \uparrow A$*

Proof. Consider the following subset of $|\uparrow A| \times |\uparrow A \triangleleft \uparrow A|$:

$$\mathcal{C} = \{(h, (h_0, h_1)) \mid \forall w \in \mathcal{2}^\omega \ h(0w) = h_0(w) \text{ and } h(1w) = h_1(w)\}$$

Let us see that it is the trace of a linear isomorphism between $\uparrow A$ and $\uparrow A \triangleleft \uparrow A$.

Firstly, \mathcal{C} clearly defines a bijection, between the webs $|\uparrow A|$ and $|\uparrow A \triangleleft \uparrow A| = |\uparrow A| \times |\uparrow A|$.

Secondly, let us see that, given $(h, (h_0, h_1))$ and $(g, (g_0, g_1))$, both in \mathcal{C} we have

$$(1) : h \frown g[\uparrow A] \iff (h_0, h_1) \frown (g_0, g_1)[\uparrow A \triangleleft \uparrow A] : (2)$$

(1) \implies (2) We assume that $h \frown g[\uparrow A]$, i.e. that $\exists w \in \mathcal{2}^\omega \ h(w) \frown g(w)$ and $\forall v < w \ h(v) = g(v)$.

In each of the two possible cases, $w = 0w'$ or $w = 1w'$, let us we show that $(h_0, h_1) \frown (g_0, g_1)[\uparrow A \triangleleft \uparrow A]$

Unless otherwise specified coherence relations are assumed to take place in the coherence space A .

0. If $w = 0w'$ we have $h_0 \frown g_0[\uparrow A]$:

- $h_0(w') \frown g_0(w')$ since $h_0(w') = h(0w') = h(w)$, $g(w) = g(0w') = g_0(w')$ and $h(w) \frown g(w)$.
- $h_0(v') = g_0(v')$ for all $v' < w'$; indeed, $0v' < 0w' = w$ hence $h_0(v') = h(0v') = g(0v') = g_0(v')$.

1. If $w = 1w'$ then $h_1 \frown g_1[\uparrow A]$ and $h_0 = g_0$:

- $h_1 \frown g_1[\uparrow A]$
 - $h_1(w') \frown g_1(w')$ since $h_1(w') = h(1w') = h(w)$, $g_1(w') = g(1w') = g(w)$, $h(w) \frown g(w)$.
 - $h_1(v') = g_1(v')$ for all $v' < w'$; indeed, $h_1(v') = h(1v') = g(1v') = g_1(v')$ since $1v' < 1w' = w$.
- $h_0 = g_0$ since $h_0(u) = h(0u) = g(0u) = g_0(u)$ because $0u < 1w' = w$.

(2) \implies (1) We assume that $(h_0, h_1) \frown (g_0, g_1)[\uparrow A \triangleleft \uparrow A]$ i.e. that either $h_0 \frown g_0[\uparrow A]$ or $(h_0 = g_0$ and $h_1 \frown g_1[\uparrow A])$. We show that in both cases we have $h \frown g[\uparrow A]$

0. If $h_0 \frown g_0[\uparrow A]$ then there exists w' such that $h_0(w') \frown g_0(w')$ and $h_0(v') = g_0(v')$ for all $v' < w'$ and $h \frown g[\uparrow A]$. Indeed:

- $h(0w') \frown g(0w')$ because $h(0w') = h_0(w')$, $g(0w') = g_0(w')$ and $h_0(w') \frown g_0(w')$.
- for all $u < 0w'$, one has $h(u) = g(u)$; indeed, if $u < 0w'$ then $u = 0u'$ with $u' < w'$, so $h(u) = h(0u') = h_0(u')$, $g(u) = g(0u') = g_0(u')$ and $h_0(u') = g_0(u')$ because $u' < w'$.

1. If $h_1 \frown g_1[\uparrow A]$ and $h_0 = g_0$ then there exists w' such that $h_1(w') \frown g_1(w')$ and $h_1(v') = g_1(v')$ for all $v' < w'$, and for all u , $h_0(u) = g_0(u)$. We have $h \frown g[\uparrow A]$:

- $h(1w') \frown g(1w')[A]$; indeed $h(1w') = h_1(w')$ and $g(1w') = g_1(w')$ and $g_1(w') \frown h_1(w')$.
- for all $v < 1w'$ one has $h(v) = g(v)$ since $v = 0u$ or $v = 1u'$ and
 - if $v = 0u$ then $h(v) = h(0u) = h_0(u) = g_0(u) = g(0u) = g(v)$.
 - if $v = 1u'$ then $u' < w'$ and therefore $h(v) = h(1u') = h_1(u')$, $h_1(u') = g_1(u')$ (because $u' < w'$), and $g(v) = g(1u') = g_1(u')$.

□

Proposition 18 (A retract of $\uparrow A$). Any coherence space A is a linear retract of the coherence space $\uparrow A$, that is there exist two linear maps as in:

$$A \begin{array}{c} \xrightarrow{f_A} \\ \circ \text{---} \circ \\ \xleftarrow{t_A} \end{array} \uparrow A$$

with $t_A \circ f_A = Id_A$ and $f_A \circ t_A \subset Id_{\uparrow A}$.

Proof. Consider $Tr(f_A) = \{(a, \underline{a}) \mid a \in |A|\} \subset |A| \times |\uparrow A|$, where $\underline{a} \in |\uparrow A|$ stands for the constant function mapping any element of 2^ω to a . Let us call $Tr(t_A)$ the symmetric of $Tr(f_A)$, that is $Tr(t_A) = \{(\underline{a}, a) \mid a \in |A|\} \subset |\uparrow A| \times |A|$. The trace $Tr(f_A)$ is the linear trace of f_A from A to $\uparrow A$ while $Tr(t_A)$ is the linear trace of t_A from $\uparrow A$ to A . The compound $f_A \circ t_A$ is $\{(\underline{a}, \underline{a}) \mid a \in |A|\}$ which is a strict subset of $Id_{\uparrow A}$, while the compound $t_A \circ f_A$ is exactly Id_A □

Proposition 19 (\uparrow is an endofunctor of Coh). Given $\ell : A \rightarrow B$ defines $\uparrow \ell : \uparrow A \rightarrow \uparrow B$ by the following trace:

$$\uparrow \ell = \{(f, g) \mid \forall w \in 2^\omega (f(w), g(w)) \in \ell\}$$

This makes \uparrow an endofunctor of the category of **Coh** of coherence spaces of definition 9.

Proof. **Firstly, let us show that $\lrcorner\ell$ defines a linear map from $\lrcorner A$ to $\lrcorner B$.** Let $(f, g), (f', g')$ be in $\lrcorner\ell$.

- Assume that $f \frown f'[\lrcorner A]$. Thus there exists w such that $f(w) \frown f'(w)[A]$ and $f(v) = f'(v)$ for all $v < w$. By definition of $\lrcorner\ell$, $(f(w), g(w))$ and $(f'(w), g'(w))$ are in ℓ which is linear, so we have $g(w) \supset g'(w)[B]$. Consider $v < w$, we have $f(v) = a = f'(v)$ and since both $(a, g(v))$ and $(a, g'(v))$ are in ℓ which is linear we have $g(v) \supset g'(v)[B]$. Applying proposition 13, there exists an u such that $g(u) \neq g'(u)$ and $g(t) = g'(t)$ for all $t < u$. We necessarily have $u \leq w$ and therefore $g(u) \frown g'(u)[B]$. Hence $g \frown g'[\lrcorner B]$.
- Assume $f = f'$. For all w , both $(f(w), g(w))$ and $(f(w), g'(w))$ are in ℓ which is linear. Therefore, for all w , one has $g(w) \supset g'(w)[B]$. Applying proposition 13 either $g = g'$ (1) or there exists an u such that $g(u) \neq g'(u)$ and $g(v) = g'(v)$ for all $v < u$ (2). When (2) holds, we have $g(u) \frown g'(u)[B]$ since $g(w) \supset g'(w)[B]$ for all w and when (1) holds we have $g = g'$. In both case we have $g \supset g'[\lrcorner B]$

It is clear that $\lrcorner Id_A = Id_{\lrcorner A}$

Let us finally show that \lrcorner commutes with linear composition. Let $\ell : A \multimap B$ and $\ell' : B \multimap C$, be two linear morphisms.

- If (f, h) is in $(\lrcorner\ell') \circ (\lrcorner\ell)$ then (f, h) is in $\lrcorner(\ell' \circ \ell)$. Indeed there exists a g in $|\lrcorner B|$ such that (f, g) is in $\lrcorner\ell$ and (g, h) is in $\lrcorner\ell'$. Thus, for all w the pair $(f(w), g(w))$ is in ℓ and the pair $(g(w), h(w))$ is in ℓ' , so that $(f(w), h(w))$ is in $\ell' \circ \ell$ for all w , i.e. (f, h) is in $\lrcorner(\ell' \circ \ell)$
- We now assume that (f, h) is in $\lrcorner(\ell' \circ \ell)$ and show that it is in $(\lrcorner\ell') \circ (\lrcorner\ell)$ too. If (f, h) is in $\lrcorner(\ell' \circ \ell)$ then $(f(w), h(w))$ is in $\ell' \circ \ell$ for all w , i.e. for all w there exists a non empty set of tokens $G(w)$ in $|B|$ with at least one $g(w) \in G(w)$ such that $(f(w), g(w))$ is in ℓ and $((g(w), h(w)))$ is in ℓ' . The point is to show that one may choose, for each w a token $g(w)$ in $G(w)$ in such a way that g is a *continuous function* from 2^ω to $|B|$ that is a token in $|\lrcorner B|$. Consider the function (f, h) from 2^ω to $|A| \times |C|$ with the discrete topology on $|A| \times |C|$. It is continuous, because the product topology of a finite product of discrete spaces is the discrete topology on the (finite) product of the involved sets. Therefore it may be described as a finite binary tree, with leaves in $|A| \times |C|$. We write it as a finite set $\{(m_i, (a_i, c_i))\}$ with the properties of generic trees (1a) and (1b) given in proposition 12. For all i , there exists w such that $f(w) = a_i$ and $h(w) = c_i$ take e.g. $w = m_i(0)^\omega$. Therefore for all i there exists b_i in $|B|$ such that (a_i, b_i) is in ℓ and (b_i, c_i) is in ℓ' , and for each i we choose one (there are finitely many i). Now, we can define $g(w)$ with the help of the property (1a) of proposition 12. For each w there exists a unique i such that $w = m_i w'$, and we define $g(w)$ to be b_i , and thus g is clearly a continuous function from 2^ω to $|B|$. Notice that the generic tree of g is not necessarily $\{(m_i, b_i)\}$ but *its normal form* according to 2 of proposition 12. Indeed, the property 1b may fail since there possibly exist i, j and m such that $m_i = m_0, m_j = m_1$ while $b_i = b_j$. Now it is easily seen that (f, g) is in $\lrcorner\ell$ and (g, h) in $\lrcorner\ell'$. Indeed, for all w there exists a unique i such that $w = m_i w'$ and we then have $f(w) = a_i, g(w) = b_i, h(w) = c_i$ and thus $(f(w), g(w)) = (a_i, b_i)$ is in ℓ and $(g(w), h(w)) = (b_i, c_i)$ is in ℓ'

□

As intuition suggests, \lrcorner is neither a monad nor a comonad. Because \lrcorner is self-dual, \lrcorner is a monad if and only if it is a comonad. The following proof adapted from [21] shows that there is no natural transformation from \lrcorner to the identity endofunctor, so \lrcorner is not a comonad.

Proposition 20. *The modality \lrcorner is neither a monad nor a comonad.*

Proof. If \lrcorner were a comonad, then there would exist a natural transformation r from the functor \lrcorner to the identity functor Id . Let $\mathbb{1}$ be the coherence space whose web is $\{*\}$ and let $\mathbb{1} \oplus \mathbb{1}$ be the coherence space whose web is $\{a, b\}$ with coherence $a \smile b$; let $\ell_a, \ell_b, \ell_{ab} : \mathbb{1} \oplus \mathbb{1} \multimap \mathbb{1}$ be the three linear morphisms from $\mathbb{1} \oplus \mathbb{1}$ to $\mathbb{1}$ respectively defined by $\ell_a = \{(a, *)\}$, $\ell_b = \{(b, *)\}$ $\ell_{ab} = \{(a, *), (b, *)\}$.

There should exist $r_{\mathbb{1} \oplus \mathbb{1}}$ from $\lrcorner(\mathbb{1} \oplus \mathbb{1})$ to $\mathbb{1} \oplus \mathbb{1}$ and $r_{\mathbb{1}}$ from $\lrcorner(\mathbb{1}) \equiv \mathbb{1}$ to $\mathbb{1}$ such that, for any x among a, b, ab :

$$\text{COM.SQ}(x): \quad r_{\mathbb{1}} \circ \lrcorner(\ell_x) = \ell_x \circ r_{\mathbb{1} \oplus \mathbb{1}}$$

For $x = ab$ one should have $r_{\mathbb{1}} \circ \lrcorner(\ell_{ab}) = \ell_{ab} \circ r_{\mathbb{1} \oplus \mathbb{1}}$ so it is mandatory that $r_{\mathbb{1}} \neq \emptyset$ so $r_{\mathbb{1}} = \text{Id}_{\mathbb{1}}$. Indeed $r_{\mathbb{1}}$ can only be $\text{Id}_{\mathbb{1}}$ or \emptyset , because there is no other linear map from $\mathbb{1}$ to itself.

Now consider the three following elements in $\lrcorner A$ (viewed as terms):

\underline{a} (constant function from 2^ω to $a \in |\mathbb{1} \oplus \mathbb{1}|$),

\underline{b} (constant function from 2^ω to $b \in |\mathbb{1} \oplus \mathbb{1}|$),

$\langle \underline{a}, \underline{b} \rangle$ (function from 2^ω mapping all $0w$ to $a \in |\mathbb{1} \oplus \mathbb{1}|$ and all $1w$ to $b \in |\mathbb{1} \oplus \mathbb{1}|$).

Observe that:

COM.SQ(a) imposes that $r_{\mathbb{1} \oplus \mathbb{1}}$ maps \underline{a} to $a \in |\mathbb{1} \oplus \mathbb{1}|$,

COM.SQ(b) imposes that $r_{\mathbb{1} \oplus \mathbb{1}}$ maps \underline{b} to $b \in |\mathbb{1} \oplus \mathbb{1}|$,

so $\langle \underline{a}, \underline{b} \rangle$ cannot be mapped by $r_{\mathbb{1} \oplus \mathbb{1}}$, hence COM.SQ(ab) cannot hold, contradiction. \square

To close the discussion on \lrcorner being or not a (co)monad, it is not difficult to be convinced, see [21] for an argument, that there also is no co-associative natural transformation from \lrcorner to $\lrcorner \lrcorner$ — although it is probably tedious to prove thoroughly

4 The Modality “Flag” in the Category of Hypercoherences

The construction is very similar, we shall be brief, and closely refer to [2]. Although we shall be brief, we think it is a good idea to recall what a hypercoherence is. The cardinal of a set M is denoted by $\#M$.

Definition 21 (Reminder on hypercoherences). A hypercoherence X is defined by its web $|X|$ (a set of tokens) and $\Gamma(X) \subset \wp_{\text{fin}}^*(X)$ i.e. a set of finite non empty subsets of the web (atomic coherence), which includes all singletons — strict atomic coherence $\Gamma^*(X)$ is $\Gamma(X)$ minus all singletons.⁵ The negation X^\perp of an hypercoherence $X = (|X|, \Gamma(X))$ is $X^\perp = (|X|, \wp_{\text{fin}}^*(X) \setminus \Gamma^*(X))$. The tensor product $X \otimes Y$ of two hypercoherences $X = (|X|, \Gamma(X))$ and $Y = (|Y|, \Gamma(Y))$ is the hypercoherence $X \times Y = (|X| \times |Y|, \Gamma(X \times Y))$ with $w \in \Gamma(X \otimes Y)$ if and only if $w_1 \in \Gamma(X)$ and $w_2 \in \Gamma(Y)$ — w_1 and w_2 are respectively the projections of w on $|X|$ and on $|Y|$ beware that in general $w \neq (w_1 \times w_2)$.

Given two hypercoherences $X = (|X|, \Gamma(X))$ and $Y = (|Y|, \Gamma(Y))$ the hypercoherence $X \multimap Y$ is defined by $X \multimap Y = (|X| \times |Y|, \Gamma(X \multimap Y))$ with $w \in \Gamma(X \multimap Y)$ if and only if (w is a finite non empty subset of $|X| \times |Y|$ and) the projections w_1 and w_2 of w on respectively $|X|$ and on $|Y|$, satisfy the following implications:

- if $w_1 \in \Gamma(X)$ then $w_2 \in \Gamma(Y)$
- if $w_1 \in \Gamma(X)$ and $\#w_1 \geq 2$ then $w_2 \in \Gamma(Y)$ and $\#w_2 \geq 2$

⁵As opposed to Girard’s qualitative domains (which include coherence spaces which are the binary generated qualitative domains)[4], a subpart of an atomic coherence is not asked to be itself an atomic coherence.

A hypercoherent subset of $|X \multimap Y|$ can be viewed as a function from X to Y .⁶

In order to define our flag modality in the category of hypercoherences, we first have to define the hypercoherence $X \triangleleft Y$ corresponding to the “before” connective from hypercoherences $|X|$ and $|Y|$:

Definition 22. Let X and Y be hypercoherences. The hypercoherence $X \triangleleft Y$ is the hypercoherence whose web is $|X| \times |Y|$ and whose strict coherence is defined by:

$$w \in \Gamma^*(X) \text{ iff } \left\{ \pi_1(w) \in \Gamma^*(X) \text{ or } [\#\pi_1(w) = 1 \text{ and } \pi_2(w) \in \Gamma^*(Y)] \right\}$$

It is easily seen that this connective is associative, self-dual, non-commutative and in between the tensor product and the par – just like in the category of coherence spaces.

Now, the self-dual modality enjoying the wanted properties is defined in the category of hypercoherences by:

Definition 23. Let X be an hypercoherence. The hypercoherence ∇X is the hypercoherence whose web is, as for coherence spaces: $|\nabla X| = \text{gt}_{|X|}$ and whose strict atomic coherence is defined by:

$$\{f_1, \dots, f_k\} \in \Gamma^*(X) \text{ iff } \exists m \in \mathcal{2}^\omega \left\{ \begin{array}{l} \{f_1(m), \dots, f_k(m)\} \in \Gamma^*(X) \\ \text{and} \\ \forall m' < m \quad \#\{f_1(m'), \dots, f_k(m')\} = 1 \end{array} \right.$$

The proofs that the hypercoherence version of ∇ enjoys the same properties as the coherence version studied in the previous section are the same *mutatis mutandis*.

Proposition 24. In the category of hypercoherences and linear maps, the modality flag as defined above, enjoys the following properties:

- flag is self-dual $(\nabla A)^\perp = \nabla(A^\perp)$
- A is a retract of (∇A)
- (∇A) is isomorphic to $(\nabla A) \triangleleft (\nabla A)$

5 Hints towards a Syntax for the “Flag” Modality

One can view the conclusion of pomset proof as a series parallel partial order of formulas (a partially ordered multiset of formulas, hence the name *pomset logic*). If the order is viewed as a set of temporal constraints $A \triangleleft B$ meaning the resource A should be consumed before the resource B is. A modal formula ∇A can be understood as repeatedly A , or A at any instant — while $!A$ rather means as many A as you wish at the instant where it is located.

A part from proof-nets, deductive systems for pomset logic are quite difficult to present, and since we do not yet have a satisfying syntax for the modality flag, we think it is wiser not to impose to the reader a technical section that does not yet yield a precise result. We nevertheless want to informally discuss what rules for the flag modality may look like.

The easiest proof system for pomset logic are proof-nets (cf. subsection 2.2). Notice that so far the only inductive definition of those proof-nets is the very complicated sequent calculus recently provided by Slavnov [27], where sequents $\vdash A_1, \dots, A_n$ are endowed with a family of binary relations R_k (with $1 \leq k \leq n$) between pairs of tuples of k conclusions $(A_{i_1}, \dots, A_{i_k})$ without common elements among A_{i_1}, \dots, A_{i_k}

⁶It is too complicated to be presented here.

— the relation R_k should stable when one applies the same permutation of the k indices to the two tuples of length k — corresponding to disjoint paths between k conclusions in a proof-net.

The calculus of structures SBV [11] is a subsystem of Pomset Logic [26] where the connective before is called “seq(ential)” and SBV is a strict subsystem of pomset logic, according to some recent work by Nguyen and Straßburger, yet unpublished [16]. This calculus consists in rewriting formulas (as terms) up to associativity of all the three connectives $\wp, \triangleleft, \otimes$ and the commutativity of the connective \wp, \otimes , starting with the axiom $\otimes_i(a_i \wp a_i^\perp)$ — some a_i may denote the same propositional variables. All the tautologies of SBV correspond to pomset proof-nets.

The contraction morphism for flag $(\wp A \triangleleft \wp A) \multimap \wp A$ is easily implemented with proof-nets. The initial two conclusions $\wp A$ and $\wp A$ are at the same place in the order (at the same instant), and a “before” link between those two conclusions as premises yields $\wp A$ instead of $\wp A \triangleleft \wp A$. The correctness criterion applies. This is easily implemented in SBV: $T[(\wp A \triangleleft \wp A)]$ rewrites to $T[\wp A]$. This quite reminiscent of the rule for contraction of $\wp A$ in linear logic, with \wp being replaced with \triangleleft .

The duplication is easy to import in a system like SBV: $\wp A \multimap (\wp A \triangleleft \wp A)$ (duplication) is interpreted as a term rewrite rule. Thus a term $T[(\wp A \triangleleft \wp A)]$ rewrites to $T[\wp A]$ and as usual in SBV this rewriting may take place within a formula. However handsome proof-nets do not handle multiple conclusions that way and consequently, without any additional structure, the rule cannot be modelled that way. What we learnt from $?$ and $!$ is that such a rule is likely to be unnecessary, because $\wp A$ is the dual of $\wp A$, so reducing a cut between two contraction suggests what the rule may look like.

The introduction yielding $\wp A$ from A is much more problematic. For $!A$ the context ought to be duplicable i.e. to be $\wp \Gamma$. Here the context ought to be replicable in time hence of the shape $\wp \Gamma$, because the dual of $\wp X$ is $\wp X$. This would require to consider \wp boxes; boxes are not fully satisfactory, but may work. When conclusions are $\wp \Gamma$ and A one could think that a box with conclusions $\wp \Gamma, \wp A$ could be introduced, but it is not that simple: an axiom $a, \neg a$ cannot be turned into $\wp a, \wp a^\perp$, because as opposed to $!$ whose dual is $?$ that can be freely introduced by the dereliction rule, the dual of \wp is the very same modality \wp . A possible solution is to make a box and to simultaneously introduce \wp on all the conclusions of the box.

An important guideline is to study how the rules that we want to design would preserve the cut-elimination property. A cut between $\wp A$ and $\wp A^\perp$ should reduce to cut(s) between A and A^\perp . When both \wp result from a box (which can be the same box, because of self-duality), or when they both come from a contraction (as in Gentzen MIX rule also known as cross-cuts), we do not yet know how those cuts should be reduced.

The proposal in [10] motivated by a computational analysis of cut elimination rather use rules like $((\wp A) \wp A) \multimap \wp A$. Such rules can be viewed as more specific, or particular cases of the ones we just discussed because $(A \triangleleft B) \multimap (A \wp B)$.

So we think it is too early to say more than the above remarks about the possible syntax of \wp .

6 Conclusion

A challenging issue is to define a deductive system enjoying cut elimination including a syntactical match of the self dual modality presented in this paper, that is a calculus for non-commutative contraction and duplication. This logical calculus could be defined as an extension of the calculus of structures with deep inference (roughly speaking, internal rewriting) [8, 11, 9, 12] or with pomset proof-nets with or without links [23, 25, 26], or with one of the sequent calculi introduced for pomset logic [18, 23, 26] not to forget the complex proposal by Sergey Slavnov [27] which is complete with respect to pomset proof-nets. The

ongoing syntactic work by Alessio Guglielmi with a self dual modality [10] looks quite appealing.

Rules may emerge from our intuition in terms of models of computation, a standard applicative domain for linear logic and related formalisms, that has been well developed for the calculus of structure and deep inference.

We think that this ancient work may give guidelines for defining such a deductive system as coherence semantics often did for linear logic. Such a logic would be a multiplicative exponential non-commutative linear logic close to classical logic, as MELL is to intuitionistic logic.

Acknowledgements: Thanks to Pierre Ageron, Thomas Ehrhard, Catherine Gourion, Thomas Streicher and especially to Jean-Yves Girard, Achim Jung and Myriam Quatrini for their constructive comments. Thanks to the reviewers of both the conference paper and the proceedings, their suggestions greatly contributed to improve the conference version as well as the proceedings version.

References

- [1] Denis Bechet, Philippe de Groote & Christian Retoré (1997): *A complete axiomatisation of the inclusion of series-parallel partial orders*. In H. Comon, editor: *Rewriting Techniques and Applications, RTA '97, LNCS 1232*, Springer Verlag, pp. 230–240, DOI: 10.1007/3-540-62950-5_74.
- [2] Thomas Ehrhard (1993): *Hypercoherences: a strongly stable model of linear logic*. *Mathematical Structures in Computer Science* 3(4), pp. 365–385, DOI: 10.1017/S0960129500000281.
- [3] Gerhard Gentzen (1934): *Untersuchungen über das logische Schließen I*. *Mathematische Zeitschrift* 39, pp. 176–210, DOI: 10.1007/BF01201353. Traduction Française de R. Feys et J. Ladrière: *Recherches sur la déduction logique*, Presses Universitaires de France, Paris, 1955.
- [4] Jean-Yves Girard (1986): *The System F of Variable Types: Fifteen Years Later*. *Theoretical Computer Science* 45, pp. 159–192, DOI: 10.1016/0304-3975(86)90044-7.
- [5] Jean-Yves Girard (1987): *Linear Logic*. *Theoretical Computer Science* 50(1), pp. 1–102, DOI: 10.1016/0304-3975(87)90045-4.
- [6] Jean-Yves Girard (1991): *A new constructive logic: classical logic*. *Mathematical Structures in Computer Science* 1(3), pp. 255–296, DOI: 10.1017/S0960129500001328.
- [7] Jean-Yves Girard (2011): *The blind spot – lectures on logic*. European Mathematical Society, DOI: 10.4171/088.
- [8] Alessio Guglielmi (1999): *A Calculus of Order and Interaction*. Technical Report WV-99-04, Dresden University of Technology.
- [9] Alessio Guglielmi (2007): *A System of Interaction and Structure*. *ACM Transactions on Computational Logic* 8(1), pp. 1–64, DOI: 10.1145/1182613.1182614.
- [10] Alessio Guglielmi (2017): *Decoupling normalization mechanisms with an eye toward concurrency*. Talk at ENS LYON seminar: programs and proofs. Slides: <http://cs.bath.ac.uk/ag/t/DNMWAETC.pdf>.
- [11] Alessio Guglielmi & Lutz Straßburger (2001): *Non-commutativity and MELL in the Calculus of Structures*. In L. Fribourg, editor: *CSL 2001, Lecture Notes in Computer Science 2142*, Springer-Verlag, pp. 54–68, DOI: 10.1007/3-540-44802-0_5.
- [12] Alessio Guglielmi & Lutz Straßburger (2011): *A System of Interaction and Structure IV: The Exponentials and Decomposition*. *ACM transaction of computational logic* 12(4), p. 23, DOI: 10.1145/1970398.1970399.
- [13] Yubao Guo & Michel Surmacs (2018): *Miscellaneous Digraph Classes*. In J. Bang-Jensen & G. Gutin, editors: *Classes of Directed Graphs*, chapter 11, Springer, pp. 517–574, DOI: 10.1007/978-3-319-71840-8_11.

- [14] Ralph Loader (1994): *Linear Logic, Totality and Full Completeness*. In: *LICS: IEEE symposium on Logic In Computer Science*, pp. 292–298, DOI: 10.1109/LICS.1994.316060.
- [15] Lê Thành Dũng Nguyễn (2019): *Proof nets through the lens of graph theory: a compilation of remarks*. Available at <https://arxiv.org/abs/1912.10606>.
- [16] Tito Nguyen & Lutz Straßburger (2021): *A complexity gap between pomset logic and system BV*. Talk at the informal workshop on Proof-Net of the GDR-I Linear Logic.
- [17] Myriam Quatrini (1995): *Sémantique cohérente des exponentielles: de la logique linéaire à la logique classique*. Thèse de Doctorat, spécialité Mathématiques, Université Aix-Marseille 2.
- [18] Christian Retoré (1993): *Réseaux et Séquents Ordonnés*. Thèse de Doctorat, spécialité Mathématiques, Université Paris 7. Available at <https://tel.archives-ouvertes.fr/tel-00585634>.
- [19] Christian Retoré (1994): *On the relation between coherence semantics and multiplicative proof nets*. Rapport de Recherche RR-2430, INRIA. Available at <https://hal.inria.fr/inria-00074245>.
- [20] Christian Retoré (1994): *A self-dual modality for "before" in the category of coherence spaces and in the category of hypercoherences*. Rapport de Recherche RR-2432, INRIA. Available at <https://hal.inria.fr/inria-00074243>.
- [21] Christian Retoré (1994): *Une modalité autoduale pour le connecteur "précède"*. In Pierre Ageron, editor: *Catégories, Algèbres, Esquisses et Néo-Esquisses*, Publications du Département de Mathématiques, Université de Caen, pp. 11–16.
- [22] Christian Retoré (1996): *Perfect matchings and series-parallel graphs: multiplicative proof nets as R&B-graphs*. In J.-Y. Girard, M. Okada & A. Scedrov, editors: *Linear'96, Electronic Notes in Theoretical Science* 3, Elsevier, pp. 167–182, DOI: 10.1016/S1571-0661(05)80416-5. [Http://www.elsevier.nl/](http://www.elsevier.nl/).
- [23] Christian Retoré (1997): *Pomset logic: a non-commutative extension of classical linear logic*. In Philippe de Groote & James Roger Hindley, editors: *Typed Lambda Calculus and Applications, TLCA'97, LNCS* 1210, pp. 300–318, DOI: 10.1007/3-540-62688-3_43.
- [24] Christian Retoré (1997): *A semantic characterisation of the correctness of a proof net*. *Mathematical Structures in Computer Science* 7(5), pp. 445–452, DOI: 10.1017/S096012959700234X.
- [25] Christian Retoré (1999): *Pomset logic as a calculus of directed cographs*. In V. M. Abrusci & C. Casadio, editors: *Dynamic Perspectives in Logic and Linguistics: Proof Theoretical Dimensions of Communication Processes, Proceedings of the 4th Roma Workshop*, Bulzoni, Roma, pp. 221–247. INRIA Research Report RR-3714 <https://hal.inria.fr/inria-00072953>.
- [26] Christian Retoré (2021): *Pomset logic: The other approach to non commutativity in logic*. In Claudia Casadio & Philip J. Scott, editors: *Joachim Lambek: on the interplay of mathematics, logic and linguistics*, Outstanding contributions to logic, Springer Verlag, pp. 299–246, DOI: 10.1007/978-3-030-66545-6.
- [27] Sergey Slavnov (2019): *On noncommutative extensions of linear logic*. *Logical Methods in Computer Science* Volume 15, Issue 3, DOI: 10.23638/LMCS-15(3:30)2019. Available at <https://lmcs.episciences.org/5774>.
- [28] Anne Sjerp Troelstra (1992): *Lectures on Linear Logic*. *CSLI Lecture Notes* 29, CSLI. (distributed by Cambridge University Press).