Pomset logic: a logical and grammatical alternative to the Lambek calculus

Christian Retoré

Pomset Logic

a Logical and Grammatical Alternative to the Lambek Calculus

Christian Retoré
LIRMM, Univ Montpellier, CNRS, Montpellier, France

January 8, 2020

Abstract

Thirty years ago, I introduced a non commutative variant of classical linear logic, called *pomset logic*, issued from a particular denotational semantics or categorical interpretation of linear logic known as coherence spaces. In addition to the multiplicative connectives of linear logic, pomset logic includes a non-commutative connective, "$<$" called *before*, which is associative and self-dual: $(A < B)^\perp = A^\perp < B^\perp$ (observe that there is no swapping), and pomset logic handles Partially Ordered MultiSETS of formulas. This classical calculus enjoys a proof net calculus, cut-elimination, denotational semantics, but had no sequent calculus, despite my many attempts and the study of closely related deductive systems like the calculus of structures. At the same period, Alain Lecomte introduced me to Lambek calculus and grammars. We defined a grammatical formalism based on pomset logic, with partial proof nets as the deductive systems for parsing-as-deduction, with a lexicon mapping words to partial proof nets. The study of pomset logic and of its grammatical applications has been out of the limelight for several years, in part because computational linguists were not too keen on proof nets. However, recently Sergey Slavnov found a sequent calculus for pomset logic, and reopened the study of pomset logic. In this paper we shall present pomset logic including both published and unpublished material. Just as for Lambek calculus, Pomset logic also is a non commutative variant of linear logic — although Lambek calculus appeared 30 years before linear logic ! — and as in Lambek calculus it may be used as a grammar. Apart from this the two calculi are quite different, but perhaps the algebraic presentation we give here, with terms and the semantic correctness criterion, is closer to Lambek's view.

1 Presentation

Lambek's *syntactic calculus* as Lambek [16] used to call his *logic*, was in keeping with his preference for algebra (confirmed with his move from categorial grammars to pregroup grammars, which are not a logical system.) Up to the invention of linear logic in the late 80s, Lambek calculus was a rather isolated logical system, despite some study of frame semantics, which are typical of substructural logics.

Linear logic [7] arose from the study of the denotational semantics of system F, itself arising from the study of ordinals. [6] For interpreting systems F (second order lambda calculus) with variable types, one needed to refine the categorical interpretation of simply typed lambda calculus with Cartesian Closed Categories. In order to quantify over types Girard considered the category of coherence spaces (first called qualitative domains) with stable maps (which preserve directed joins and pullbacks). A finer study of coherence spaces led Girard to discompose the arrow type construction in to two steps: one is to contract several object of type A into one (modality/exponential !) and the other one being linear implication (noted $\multimap$) which rather corresponds to a change of state than to a consequence relation.

Linear logic was first viewed as a proof system (sequent calculus or proof nets) which is well interpreted by coherence spaces. The initial article [7] also included the definition of phase semantics, that resembles frame semantics developed for the Lambek calculus. It was not long before the connection between linear logic and Lambek calculus was found: after some early remarks by Girard, Yetter [50] observed the connection at the semantic level, while Abrusci [1] explored the syntactic, proof theoretical connection, while [34] explored proof nets and completed the insight of [46]. Basically Lambek calculus is non commutative intuitionistic multiplicative logic, the order between the two restrictions, intuitionistic and non commutative, being independent. An important remark, that I discussed with Lamarche in [15], says that non commutativity requires linearity in order to get a proper logical calculus.

Around 1988, my PhD advisor Jean-Yves Girard pointed to my attention a binary non commutative connective $<$ in coherence spaces. In coherence spaces, this connective has intriguing properties:

- $<$ is self dual $(A < B)^\perp \equiv (A^\perp < B^\perp)$, without swapping the two components — by $X \equiv Y$ we mean that there is a pair of canonical invertible linear maps between X and Y.
- $<$ is non commutative $(A < B) \not\equiv (B < A)$
- $<$ is associative $((A < B) < C) \equiv (A < (B < C))$;
- it lies in between the commutative conjunction $\otimes$ and disjunction $\parr$ there is a canonical linear map from $A(\otimes B)$ to $(A < B)$ an one from $(A < B)$ to $(A \parr B)$[1];

I designed a proof net calculus with this connective, in which a sequent, that is the conclusion of a proof, is a partially ordered multiset of formulas. This proof net calculus enjoys cut-elimination and a sound and faithful (coherence) semantics [2] preserved under cut elimination. I proposed a version of sequent

[1]Coherence spaces validate the MIX rule $(A \otimes B) \multimap (A \parr B)$

[2]In the sense that having an interpretation is the same as being syntactically correct cf. theorem 8

calculus that easily translates into those proof nets and enjoys cut-elimination as well. [32] However despite many attempts by me and others (Sylvain Pogodalla, Lutz Straßburger) over many years we did not find a sequent calculus that would be complete w.r.t. the proof nets. Later on, Alessio Guglielmi, soon joined by Lutz Straßburger, designed the calculus of structures, a term calculus more flexible than sequent calculus (deep inference) with the *before* connective [9, 10, 11], a system that is quite close to dicograph rewriting [40, 41], described in section 3.2. They tried to prove that one of their systems called BV was equivalent to pomset logic and they did not succeed. As a reviewer of my habilitation [42] Lambek wrote:

> *He constructs a model of linear logic using graphs, which is new to me. His most original contribution is probably the new binary connective which he has added to his non commutative version of linear logic, although I did not find where it is treated in the sequent calculus. (J. Lambek, Dec. 3, 2001)*

I deliberately omitted my work on sequent calculus in my habilitation manuscript, because none of the sequent calculi I experimented with was complete w.r.t. pomset proof nets which are "perfect", i.e. enjoy all the expected proof theoretical properties. In addition, by that time, I did not yet have a counter example to my proposal of a sequent calculus, the one in picture 5 of section 6 was found ten years later with Lutz Straßburger.

However, very recently, Sergey Slavnov found a sequent calculus that is complete w.r.t. pomset proof nets. [48] The structure of the decorated sequents that Slavnov uses is rather complex [3] and the connective $<$ is viewed as the identification of two dual connectives one being more like a $\otimes$ and the other more like a $\invamp$. As this work is not mine I shall not say much about it, but Slavnov's work really sheds new light on pomset logic. Given the complexity of this sequent calculus it is pleasant to have some simple sequent calculus and a rewriting system for describing most useful proof nets e.g. the one used for grammatical purposes.

Pomset logic and the Lambek calculus systems share some properties:

- They both are linear calculi;
- They both handle non commutative connective(s) and structured sequents;
- They both have a sequent calculus;
- They both enjoy cut-elimination;
- They both have a complete sequent calculus (regarding pomset logic the complete sequent calculus is quite new);

[3] A decorated sequent according to Slavnov is a multiset of pomset formulas $A_1, \ldots, A_n$ with $p \leq n/2$ binary relations $(R_k)1 \leq k \leq p$ between sequences of length $p \leq n/2$ of formulas from Γ; those relations are such that whenever $(B_1, \ldots, B_k R_k(C_1, \ldots, C_k)$ the two sequences $(B_1, \ldots, B_k)$ and $(C_1, \ldots, C_k)$ have no common elements and $(B_1, \ldots, B_k)R_k(C_1, \ldots, C_k)$ entails $(B_{\sigma(1)}, \ldots, B_{\sigma(k)})R_k(C_{\sigma(1)}, \ldots, C_{\sigma(k)})$ for any permutation σ of $\{1, \ldots, k\}$ – those relations correspond top the existence of disjoint paths in the proof nets from B_i to C_i.

- They both can be used as a grammatical system.

However Lambek calculus and pomset logic are quite different in many respects:

- Lambek calculus is naturally an intuitionistic calculus while pomset logic is naturally a classical calculus — although in both cases variants of the other kind can be defined.

- Lambek calculus is a restriction of the usual multiplicative linear logic according to which the connectives are no longer commutative, while pomset logic is an extension of usual commutative multiplicative linear logic with a non commutative connective.

- Lambek calculus deals with totally ordered multisets of hypotheses while pomset logic deals with partially ordered multisets of formulas. As grammatical systems, pomset logic allows relatively free word order, while Lambek calculus only deals with linear word orders.

- Lambek calculus has an elegant truth-value interpretation within the subsets of a monoid (frame semantics, phase semantics), while there is not such a notion for pomset logic.

- Lambek calculus has no simple concrete interpretation of proofs up to cut elimination (denotational semantics) while coherence semantics faithfully interprets the proofs of pomset logic.

This list shows that those two comparable systems also have many differences. However, the presentation of Pomset logic provided by the present article make Lambek calculus and pomset logic rather close on an abstract level. As he told many of us, Lambek did not like standard graphical or geometrical presentation of linear logic like proof nets. He told me several times that moving from geometry to algebra has been a great progress in mathematics and solved many issues, notably in geometry, and that proof net study was going the other way round. I guess this is related to what he said about theorem 8.

> *It seems that this ingenious argument avoids the complicated long trip condition of Girard. It constitutes a significant original contribution to the subject. (J. Lambek,Dec 3 2001)*

This paper is a mix (!) of easy to access published work, [5, 34, 4, 37, 38, 35, 43] research reports and more confidential publications [32, 31, 5, 18, 34, 31, 33, 18, 19, 15, 20, 40, 41, 39, 42, 30] unpublished material between 1990 and 2020, that are all presented in the same and rather new unified perspective; the presented material can be divided into three topics:

proof nets handsome proof nets both for MLL Lambek calculus and pomset logic, and other work on proof nets [5, 15, 34],

combinatorics (di)cographs and SP orders [31, 32, 35, 4, 40, 41, 39, 42, 43, 42, 43, 30],

coherence semantics [32, 33, 38, 42],

grammatical applications of pomset logic to computational linguistics [18, 19, 20, 40, 41, 42].

The contents of the present article is divided into six sections as follows:

2. We first present results on series parallel partial orders, cographs and dicographs that subsumes those two notions and present dicograph either as SP pomset of formulas or as dicographs of atoms, and explain the guidelines for finding a sequent calculus. This combinatorial part is a prerequisite for the subsequent sections.

3. We then present proofs in an algebraic manner, with deduction rules as term rewriting.

4. Proof nets without links, the so called handsome proof nets, are presented as well as the cut elimination for them.

5. The semantics of proof nets, preserved under cut elimination and equivalent to their syntactic correctness is then presented.

6. Then the sequentialisation "the quest" of a complete sequent calculus is discussed and we provide an example of a proof net that does not derive from any simple sequent calculus.

7. Finally we explain how one can design grammars by associating words with partial proof nets of pomset logic.

2 Structured sequents as dicographs of formulas

2.1 Looking for structured sequents

The formulas we consider are defined from atoms (propositional variables or their negation) by means of the usual commutative multiplicative connectives $\wp$ and $\otimes$ together with the new non commutative connective $<$ (before)— the three of them are associative.

It is assumed that formulas are always in negative normal form: negation only apply to propositional variables; this is possible and standard when negation is involutive and satisfies the De Morgan laws:

$$\begin{array}{rcl} (A^{\perp})^{\perp} & = & A \\ (A \wp B)^{\perp} & = & (A^{\perp} \otimes B^{\perp}) \\ (A < B)^{\perp} & = & (A^{\perp} < B^{\perp}) \\ (A \otimes B)^{\perp} & = & (A^{\perp} \wp B^{\perp}) \end{array}$$

We want to deal with series parallel partial orders of formulas: $O_1 \wp O_2$ corresponds to parallel composition of partial orders (disjoint union) and $O_1 < O_2$ corresponds to the series composition of partial orders (every formula in the first partial order O_1 is lesser than every formula in the second partial order O_2). Thus, a formula written with $\wp$ and $<$ corresponds to a partial order between its atoms. Unsurprisingly, we firstly need to study a bit partial orders defined with series and parallel composition.

However, what about the multiplciative, conjunction namely the $\otimes$ connecitve? It is commutative, but it is distinct from $\wp$. In order to include $\otimes$ in this view, where formulas are binary relations on their atoms, we consider, the more general class of irreflexive binary relations that are obtained by $\wp$ parallel composition, $<$ series composition and $\otimes$ symmetric series compositions, which basically consists in adding the relations of $R_1 < R_2$ and the ones of $R_2 < R_1$. The relations that are defined using $\wp$, $\otimes$, $<$ are called directed cographs or dicographs for short.

If only $\wp$ and $\otimes$ are used the relations obtained are cographs. They have already been quite useful for studying MLL, see e.g. theorem 4 thereafter.

Before defining pomset logic, we need a presentation of directed cographs.

2.2 Directed cographs or dicographs

An irreflexive relation $R \subset P^2$ may be viewed as a graph with vertices P and with both directed edges and undirected edges but without loops. Given an irreflexive relation R let us call its directed part (its arcs) $\vec{R} = \{(a,b) \in R | (b,a) \notin R\}$ and its symmetric part (its edges) $\bar{R} = \{(a,b) \in R | (b,a) \in R\}$. It is convenient to note a–b for the edge or pair of arcs $(a,b),(b,a)$ in $\bar{R}$ and to denote $a \to b$ for (a,b) in R when (b,a) is not in R.

We consider the class of dicographs, dicographs for short, which is the smallest class of binary irreflexive relations containing the empty relation on the singleton sets and closed under the following operations defined on two cographs with disjoint domains E_1 and E_2 yielding a binary relation on $E_1 \uplus E_2$

- symmetric series composition $R_1 \mathbin{\widehat{\otimes}} R_2 = R_1 \uplus R_2 \uplus (E_1 \times E_2) \uplus (E_2 \times E_1)$
- directed series composition $R_1 \mathbin{\widehat{<}} R_2 = R_1 \uplus R_2 \uplus (E_1 \times E_2)$
- parallel composition $R_1 \widehat{\wp} R_2 = R_1 \uplus R_2$

Whenever there are no directed edges (a.k.a. arcs) the dicograph is a cograph ($\widehat{<}$ is not used). Cographs are characterised by the absence of P_4 as many people (re)discovered including us [35], see e.g. [14].

Whenever there are only directed edges (a.k.a. arcs) the dicograph is an SP order ($\widehat{\otimes}$ is not used) — as rediscovered in [32], see e.g. [24]

Let us call this class the class of *dicographs.*

We characterised the class of directed dicographs as follows [4, 40, 41]:

Theorem 1 *An irreflexive binary relation R is a dicographs if and only if:*

- *$\vec{R}$ is N-free ($\vec{R}$ is an* SP *order).*
- *$\bar{R}$ is P_4-free ($\bar{R}$ is a cograph).*
- *Weak transitivity:*
  forall a, b, c in the domain of R
  if $(a,b) \in \vec{R}$ and $(b,c) \in R$ then $(a,c) \in R$ and
  if $(a,b) \in R$ and $(b,c) \in \vec{R}$ then $(a,c) \in R$

A dicograph can be described with a term in which each element of the domain appears exactly once. This term is written with the three binary operators $\widehat{\otimes}$, $\widehat{\wp}$ and $\widehat{<}$ and for a given dicograph this term is unique up to the associativity of the three operators, and to the commutativity of the first two, namely $\widehat{\wp}$ and $\widehat{\otimes}$.

The dual $R^\perp$ of a dicograph R on P is defined as follows: points are given a $\perp$ superscript, $\vec{R}^\perp = \vec{R}$ and $(\overline{R^\perp}) = (P^2 \setminus \bar{R}) \setminus \{(x,x) | x \in P\}$ or $(a^\perp) = (a)^\perp$, $(a^\perp)^\perp = a$, $(X \mathbin{\widehat{\otimes}} Y)^\perp = (X^\perp \widehat{\wp} Y^\perp)$, $(X \widehat{\wp} Y)^\perp = (X^\perp \mathbin{\widehat{\otimes}} Y^\perp)$, $(X \mathbin{\widehat{<}} Y)^\perp = (X^\perp \widehat{\wp} Y^\perp)$.

Two points a and b of P are said to be equivalent w.r.t. a relation whenever for all $x \in P$ with $x \neq a, b$ one as $(x,a) \in R \Leftrightarrow (x,b) \in R$ and $(a,x) \in R \Leftrightarrow (b,x) \in R$. There are three kinds of equivalent points:

- Two points a and b in a dicograph are said to be freely equivalent in a dicograph (notation $a \stackrel{..}{\sim} b$) whenever the term can be written (using associativity of $\widehat{\wp}$ and $\widehat{<}$ and the commutativity of $\widehat{\wp}$) $T[a \widehat{\wp} b]$. In other words, $a \sim b$, $(a,b) \notin R$, $(b,a) \notin R$.
- Two points a and b in a dicograph are said to be arc equivalent in a dicograph (notation $a \stackrel{\rightarrow}{\sim} b$) whenever the term can be written (using associativity of $\widehat{\wp}$, $\widehat{\otimes}$ and $\widehat{<}$ and the commutativity of $\widehat{\wp}$ and $\widehat{\otimes}$) $T[a \mathbin{\widehat{<}} b]$. In other words, $a \sim b$, $(a,b) \in R$, $(b,a) \notin R$.
- Two points a and b in a dicograph are said to be edge equivalent in a dicograph (notation $a \stackrel{-}{\sim} b$) whenever the term can be written (using associativity of $\widehat{\wp}$ and $befsp$ and the commutativity of $\widehat{\wp}$) $T[a \mathbin{\widehat{\otimes}} b]$. In other words, $a \sim b$, $(a,b) \in R$, $(b,a) \in R$.

2.3 Dicograph inclusion and (un)folding

The order on a multiset of formulas, can be viewed as a set of contraints. Hence, when a sequent is derivable with an SP order I it is also derivable with a sub SP order $J \subset I$ — we named this structural rule entropy [32]. Most of the transformations of a dicograph into a smaller (w.r.t. inclusion) dicograph preserve provability. Hence we need to characterise the inclusion of a dicograph into another and possibly to view the inclusion as a computational process that can be performed step by step. Fortunately,in [4] we characterised the inclusion of a dicograph in another dicograph by a rewriting relation:

	rule name	*dicograph*	$\rightsquigarrow$	*dicograph′*
$\times$	$\widehat{\otimes}\widehat{\parr}4$	$(X \mathbin{\widehat{\parr}} Y) \mathbin{\widehat{\otimes}} (U \mathbin{\widehat{\parr}} V)$	$\rightsquigarrow$	$(X \mathbin{\widehat{\otimes}} U) \mathbin{\widehat{\parr}} (Y \mathbin{\widehat{\otimes}} V)$
	$\otimes\parr 3$	$(X \mathbin{\widehat{\parr}} Y) \mathbin{\widehat{\otimes}} U$	$\rightsquigarrow$	$(X \mathbin{\widehat{\otimes}} U) \mathbin{\widehat{\parr}} Y$
	$\otimes\parr 2$	$Y \mathbin{\widehat{\otimes}} U$	$\rightsquigarrow$	$U \mathbin{\widehat{\parr}} Y$
	$\otimes{<}4$	$(X \mathbin{\widehat{<}} Y) \mathbin{\widehat{\otimes}} (U \mathbin{\widehat{<}} V)$	$\rightsquigarrow$	$(X \mathbin{\widehat{\otimes}} U) \mathbin{\widehat{<}} (Y \mathbin{\widehat{\otimes}} V)$
	$\otimes{<}3l$	$(X \mathbin{\widehat{<}} Y) \mathbin{\widehat{\otimes}} U$	$\rightsquigarrow$	$(X \mathbin{\widehat{\otimes}} U) \mathbin{\widehat{<}} Y$
	$\otimes{<}3r$	$Y \mathbin{\widehat{\otimes}} (U \mathbin{\widehat{<}} V)$	$\rightsquigarrow$	$U \mathbin{\widehat{<}} (Y \mathbin{\widehat{\otimes}} V)$
	$\otimes{<}2$	$Y \mathbin{\widehat{\otimes}} U$	$\rightsquigarrow$	$U \mathbin{\widehat{<}} Y$
	${<}\parr 4$	$(X \mathbin{\widehat{\parr}} Y) \mathbin{\widehat{<}} (U \mathbin{\widehat{\parr}} V)$	$\rightsquigarrow$	$(X \mathbin{\widehat{<}} U) \mathbin{\widehat{\parr}} (Y \mathbin{\widehat{<}} V)$
	${<}\parr 3l$	$(X \mathbin{\widehat{\parr}} Y) \mathbin{\widehat{<}} U$	$\rightsquigarrow$	$(X \mathbin{\widehat{<}} U) \mathbin{\widehat{\parr}} Y$
	${<}\parr 3r$	$Y \mathbin{\widehat{<}} (U \mathbin{\widehat{\parr}} V)$	$\rightsquigarrow$	$U \mathbin{\widehat{\parr}} (Y \mathbin{\widehat{<}} V)$
	${<}\parr 2$	$Y \mathbin{\widehat{<}} U$	$\rightsquigarrow$	$U \mathbin{\widehat{\parr}} Y$

Figure 1: A complete rewriting system for dicograph inclusion. Beware that the first rule $\widehat{\otimes}\widehat{\parr}4$ marked with a $\times$ is wrong when the rewriting rule is viewed as a linear implication on formulas: $(X\widehat{\parr}Y)\mathbin{\widehat{\otimes}}(U\widehat{\parr}V) \not\multimap (X\mathbin{\widehat{\otimes}}U)\widehat{\parr}(Y\widehat{\parr}V)$ although all other rewriting rules are correct when viewed as linear implications.

Theorem 2 *A dicograph R' is included into a dicograph R if and only if the term R rewrites to the term R' using the rules of figure 2.3 up to the associativity of $\widehat{\otimes}$, $\widehat{<}$ and $\widehat{\parr}$, and to the commutativity of $\widehat{\parr}$ and $\widehat{\otimes}$.*

2.4 Folding and unfolding pomset logic sequents

A structured sequent of pomset logic (resp. of MLL) is a multiset of formulas of pomset logic (resp. of MLL) with the connectives $<, \otimes, \parr$ endowed with a dicograph.

On such sequents one may define "folding" and "unfolding" which transform a dicograph of formulas into another dicograph of formulas by combining two equivalent formulas A and B of the dicograph into one formula $A * B$ (folding) or by splitting one compound formula $A * B$ into its two immediate subformulas A and B with A and B equivalent in the dicograph. More formally:

Folding Given a multiset of formulas $X_1, \ldots, X_n$ endowed with a dicograph T,

- $\parr$ if $X_i \overset{..}{\sim} X_j$ in T rewrite $T[X_i \widehat{\parr} X_j]$ into $T[(X_i \parr X_j)]$ — in the multiset, the two formulas X_i and X_j have been replaced with a single $X_i \parr X_j$.
- $<$ if $X_i \overset{\rightarrow}{\sim} X_j$ in T rewrite $T[X_i \widehat{<} X_j]$ into $T[(X_i < X_j)]$ — in the multiset, the two formulas X_i and X_j have been replaced with a single formula $X_i < X_j$.
- $\otimes$ if $X_i \bar{\sim} X_j$ in T rewrite $T[X_i \widehat{\otimes} X_j]$ into $T[(X_i \otimes X_j)]$ — in the multiset, the two formulas X_i and X_j have been replaced with a single formula $X_i \otimes X_j$.

Unfolding is the opposite:

- $\parr$ turn $T[(X_i \parr X_j)]$ into $T[X_i \widehat{\parr} X_j]$ — in the multiset, the formula $X_i \parr X_j$ has been replaced with two formulas X_i and X_j with $X_i \overset{..}{\sim} X_j$
- $<$ turn $T[(X_i < X_j)]$ into $T[X_i \widehat{<} X_j]$ — in the multiset, the formula $X_i \parr X_j$ has been replaced with two formulas X_i and X_j with $X_i \overset{\rightarrow}{\sim} X_j$
- $\otimes$ turn $T[(X_i \otimes X_j)]$ into $T[X_i \widehat{\otimes} X_j]$ — in the multiset, the formula $X_i \parr X_j$ has been replaced with two formulas X_i and X_j with $X_i \overset{\rightarrow}{\sim} X_j$

2.5 A sequent calculus attempt with SP pomset of formulas

If we want to extend multiplicative linear logic with a non commutative multiplicative self dual connective (rather than to restrict existing connective to be non commutative), and want to handle partially ordered multisets of formulas, with $A < B$ corresponding to "the subformula A is smaller than the subformula B".

That way one may think of an order on computations: a cut between $(A < B)^\perp$ and $A^\perp < B^\perp$ reduces to two smaller cuts $A{-}cut{-}A^\perp$ and $B{-}cut{-}B^\perp$ with the cut on A being prior to the cut on B, while a cut between $(A \parr B)^\perp$ and $A^\perp \otimes B^\perp$ reduces to two smaller cuts $A{-}cut{-}A^\perp$ and $B{-}cut{-}B^\perp$ with the cut on A being in parallel with the cut on B. This makes sense when linear logic proofs are viewed as programs and cut-elimination as computation.

Doing so one may obtain a sequent calculus using partially ordered multisets of formulas as in [32] but if one wants a sequent with several conclusions that are partially ordered to be equivalent to a sequent with a unique conclusion, one has to only consider SP partial orders of formulas, as defined in subsection 2.2 with parallel composition noted $\widehat{\parr}$ and series composition noted $\widehat{<}$.

If we want all formulas in the sequent to be ordered the calculus should handle right handed sequents i.e. be classical.[4]

As seen above, we can represent this SP partially ordered multiset of formulas endowed with an SP order by an SP term whose points are the formulas and such

[4] Lambek calculus is intuitionistic and when it is turned into a classical systems, formulas are endowed with a cyclic order,[50, 1, 15], i.e. a ternary relation which is not an order and which is quite complicated when partial — see the "seaweeds" in [3].

a term is unique up to the commutativity of $\widehat{\wp}$ and the associativity of $\widehat{\wp}$ and $\widehat{<}$.

We know how the tensor rule and the cut rule behave w.r.t. formulas. The only aspect that deserves some tuning, is the order on the formulas after applying those binary rules. Our choice is guided by two independent criteria:

1. The resulting partial order should be an SP order.

2. This sequent calculus should enjoy cut elimination:

   - If there is a cut between $A \otimes B$ and $A^\perp \wp B^\perp$ with $A \otimes B$ coming immediately from a $\otimes$ rule from $\vdash \Gamma, A$ and $\vdash \Delta, B$ and $A^\perp \wp B^\perp$ coming immediately from a $\wp$ rule from $\vdash \Theta, A^\perp, B^\perp$ one should be able to locally turn those rules into two consecutive cuts, one between Θ, A, B and $\vdash \Gamma, A$ and then one with $\vdash \Delta, B$.
   - If there is a cut between $A < B$ and $A^\perp < B^\perp$ with $A < B$ coming immediately from a $<$ rule from $\vdash \Gamma, A, B$ and $A^\perp < B^\perp$ coming immediately from a $<$ rule from $\vdash \Theta, A^\perp, B^\perp$ one should be able to locally turn those rules into two consecutive cuts, one between Θ, A, B and $\vdash \Gamma, A$ and then one with $\vdash \Delta, B$.

A simple sequent calculus is provided in figure 2. [5]

A property of this calculus is that cuts can be part of the order on conclusions. Indeed, one may define a cut as a formula $K \otimes K^\perp$ that never is used as a premise of a logical rules. That way, the order can be viewed as an order on computation. A cut $\Gamma[((A \otimes B) \otimes (A^\perp \wp B^\perp)]$ reduces into two cuts that are $\ddot{\sim}$: $\Gamma[((A \otimes A^\perp) \widehat{\wp} (B \otimes B^\perp)]$, while cut $\Gamma[((A < B) \otimes (A^\perp < B^\perp)]$ reduces into two cuts that are $\vec{\sim}$: $\Gamma[((A \otimes A^\perp) \mathbin{\widehat{<}} (B \otimes B^\perp)]$ — beware that $K \otimes K^\perp$ is a cut and that a $\widehat{*}$ operation on dicographs is different from the $*$ connective, which combines formulas. When one of the two premises of the cut is an axiom, this axiom and the cut are simply removed from the proof as usual, and this is possible because cut only applies when the two premises are isolated in the SP order. An alternative proof of cut elimination can be obtained from the cut elimination theorem for proof nets with links or without to be defined in sections 3.2 or 7.1, as established in [32, 37, 41] — because the reduction of a proof net coming from a sequent calculus proof also comes from a sequent calculus proof. Thus one has:

Theorem 3 *Sequent calculus of figure 2 enjoys cut-elimination.*

[5] An alternative solution to have on SP orders is to have $\otimes$ rule between two minimum in their order component, and to have cut between two formulas one of which is isolated in its ordered sequent. This alternative is trickier and up to our recent investigation does not enjoy better properties than the version given above in figure 2

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma \mathbin{\widehat{\lessgtr}} \Delta}\, dimix \qquad\qquad \frac{\vdash \Gamma}{\vdash \Gamma'}\ \text{entropy}(\Gamma' \text{ sub sp order of } \Gamma)$$

$$\overline{\vdash a, a^\perp}$$

$$\frac{\vdash A \widehat{\parr} \Gamma \qquad \vdash B \widehat{\parr} \Delta}{\vdash \Gamma \widehat{\parr} (A \otimes B) \widehat{\parr} \Delta}\ \otimes \text{ / cut when } A = B^\perp$$

$$\frac{\vdash \Gamma[A \widehat{\parr} B]}{\vdash \Gamma[A \parr B]}\ \parr \text{ when } A \mathrel{\ddot{\sim}} B \qquad\qquad \frac{\vdash \Gamma[A \mathbin{\widehat{\lessgtr}} B]}{\vdash \Gamma[A < B]} < \text{ when } A \mathrel{\overset{\rightarrow}{\sim}} B$$

Figure 2: Sequent calculus on SP pomset or formulas; called SP-pomset sequent calculus

3 Pomset logic and MLL as (di)cograph rewriting:

Before we define a deductive system for pomset logic, let us revisit (as we did in [39, 43]) the deductive system of Multiplicative Linear Logic (MLL). Those results are highly inspired from proof nets, but once they are established they can be presented before proof nets are defined.

In this section a sequent is simply a dicograph of *atoms* which as explained above can be viewed using *folding* of section 2.4 as a dicograph of formulas or as an SP order between formulas depending on how many *folding* transformations and which one are performed.

Regarding, multiplicative linear logic (MLL), observe that $\text{AX}_n = \bigotimes_{1 \leq i \leq n} (a_i \parr {a_i}^\perp)$ is the largest cograph or even the largest dicograph w.r.t. inclusion that could be derived in MLL — indeed there cannot be any tensor link nor any before connection between the two dual occurrences of atoms issued from the same axiom link, for two reasons: first in sequent calculus they cannot lie in different sequents and therefore they cannot be conjoined by $<$ or $\otimes$; second, as explained in subsection 3.2, in the proof net this would result in a prohibited (AE) cycle. Observe that $\bigotimes_i (a_i \parr {a_i}^\perp)$, the largest derivable cograph in MLL is acutally derivable in MLL, hence in any extension of MLL:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash a_1, {a_1}^{\perp}}{\vdash {\scriptstyle AX}_1 : a_1 \parr {a_1}^{\perp}}\parr \quad \dfrac{\vdash a_2, {a_2}^{\perp}}{\vdash a_2 \parr {a_2}^{\perp}}\parr}{\vdash {\scriptstyle AX}_2 : \otimes_{1 \leq i \leq 2}(a_i \parr {a_i}^{\perp})}\otimes \quad \dfrac{\vdash a_3, {a_3}^{\perp}}{\vdash a_3 \parr {a_3}^{\perp}}\parr}{\vdash {\scriptstyle AX}_3 : \otimes_{1 \leq i \leq 3}(a_i \parr {a_i}^{\perp})}\otimes \quad \dfrac{\vdash a_4, {a_4}^{\perp}}{\vdash a_4 \parr {a_4}^{\perp}}\parr}{\vdash {\scriptstyle AX}_4 : \otimes_{1 \leq i \leq 4}(a_i \parr {a_i}^{\perp}) \qquad \cdots}\otimes}{\vdash {\scriptstyle AX}_5 : \cdots}\otimes
$$

3.1 Standard multiplicative linear logic as cograph rewriting

In [39] we considered an alternative way to derive theorems of usual multiplicative linear logic MLL, by considering a formula as a binary relation, and more precisely as a cograph over its atoms, by viewing $\widehat{\otimes}$ as $\otimes$ and $\widehat{\parr}$ as $\parr$. As there is no $<$ connective in linear logic the series composition is not used, and there is no SP order on conclusions.

Because of the chapeau of the present section 3 any sequent of MLL can be viewed is a cograph $C[a_1, a_1^{\perp}, a_2, a_2^{\perp}, \ldots a_n, a_n^{\perp}]$ on $2n$ atoms that is included into AX_n. Because of theorem 2.3, AX_n rewrites to $C[a_1, a_1^{\perp}, a_2, a_2^{\perp}, \ldots a_n, a_n^{\perp}]$ using the rules of figure 2.3 that concern $\parr$ and $\otimes$ i.e. $\otimes\parr 4$, $\otimes\parr 3$ and $\otimes\parr 2$. Observe that when viewed as a linear implication (considering the rules involving those two connectives), the first line $\otimes\parr 4$ is an incorrect linear implication, while $\otimes\parr 3$ is derivable in MLL and $\otimes\parr 2$ in MLL+MIX where the rule MIX is the one studied in [5], which also is derivable with $\otimes\parr 2$:

$$
\dfrac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma, \Delta}\,\text{MIX}
$$

Actually *all* tautologies of multiplicative linear logic MLL can be derived using $\otimes\parr 3$ from an axiom $AX_n = \bigotimes_{1 \leq i \leq n}(a_i \parr {a_i}^{\perp})$, and *all* tautologies of linear logic enriched with the MIX rule, MLL+MIX, can be derived by $\otimes\parr 3$ and $\otimes\parr 2$ (MIX).

Thus, we can define a proof system gMLL for MLL working with sequents as cographs of atoms as follows. Axioms are AX_n: $\widehat{\otimes}_i\ (a_i \widehat{\parr} {a_i}^{\perp})$ (the two dual atoms are connected by an edge in a different relation called A for A axioms). There is just one deduction rule presented as a rewrite rule (up to commutativity and associativity): $\otimes\parr 3$.

Let us call this deductive system gMLL (g for graph), then [39, 43] established that cograph rewriting is an alternative proof systems to MLL and MLL+MIX.

Theorem 4 *MLL proves a sequent $\vdash \Gamma$ with $2n$ atoms if and only if gMLL proves the unfolding Γ^{cog} of Γ (the cograph Γ^{cog} of atoms corresponding to Γ,* *that is the $\parr$ of the unforging of each formula in* Γ*), i.e.* AX_n *rewrites to* Γ^{cog} *using* $\otimes\parr 3$.

MLL+MIX proves a sequent $\vdash \Gamma$ *with* $2n$ *atoms if and only if gMLL+mix proves the unfolding* Γ^{cog} *of* Γ*, i.e.* AX_n *rewrites to* Γ^{cog} *using* $\otimes\parr 3$ *and* $\otimes\parr 2$.

Proof. Easy induction on sequent calculus proofs see e.g. [39, 43]. A direct proof by Straßbruger can be found in [49]. □

The interesting thing is that all proofs can be transformed that way. Unfortunately it if much easier with an *inductive* definition of proofs like sequent calculus, and, unfortunately for pomset logic, it is hard to prove it directly on a non inductive notion of proof like proof nets.

Proposition 1 *The calculi gMLL and gMLL+mix can safely be extended to structured sequents of formulas of MLL (not just atoms), i.e. cographs of MLL formulas with the rules of folding and unfolding with the same results.*

Proof. This is just an easy remark, based on proof nets, which can be viewed as a consequence subsection 7.1. □

3.2 Pomset logic as a calculus of dicographs: DICOG-RS

Using the above results for MLL suggests defining a deductive system for pomset logic in the same manner. All rewriting rules are correct but $\otimes\parr 4$: they correspond to proof nets or to sequent calculus derivations (with the SP-pomset sequent calculus of figure 2) and to canonical linear maps in coherence spaces. So it suggest that a rewriting system defined as gMLL+mix in the previous section (but with dicographs instead of cographs) might yield all the proofs we want e.g. all correct proof nets.

Axioms $AX_n = \widehat{\otimes}_{1\leq i\leq n}(a_i \widehat{\parr} {a_i}^{\perp})$ is a tautology.

Rules Whenever a dicograph of atoms D which is a tautology rewrites to a dicograph D' (hence with the sames atoms) by any of the 10 rules $\otimes\parr 3$, $\otimes\parr 2$, $\otimes < 4$, $\otimes < 3l$, $\otimes < 3r$, $\otimes < 2$, $<\parr 4$, $<\parr 3l$, $<\parr 3r$, $<\parr 2$ of figure 2.3 — i.e. all rules of figure 2.3 but $\otimes\parr 4$.

Unfortunately, proving that all proof nets are derivable by rewriting is not simpler than proving that they can be obtained from the sequent calculus. This would entail the equivalence of pomset logic with BV calculus as discussed in [12].

3.3 Cuts

What about the cut rule? For such logical systems based on rewriting systems like gMLL(+MIX), of the DICOG-RS view of pomset logic, which does not work with "logical rules" in the standard sense, there are no binary rules that would combine a K and a $K^{\perp}$. So the only view of a cut is simply a tensor $K \otimes K^{\perp}$

which never is inserted inside a $\otimes$ formula. A dicograph may be written $D = R[b_1, b_1^\perp, \ldots, b_n, b_n^\perp, K[c_1, \ldots c_k] \otimes K^\perp[c_1^\perp; \ldots c_k^\perp]]$. Observe that $K^\perp$ contains the duals of the atoms in K, because it is a cut, and that there is one $b_i^\perp$ for each b_i, because they are the atoms of D minus the well balanced atoms of K and $K^\perp$, one cannot say that the pair $b_i, b_i^\perp$ corresponds to some $a_i \parr a_i^\perp$ from AX_n — necessarily for some pair $a_i, a_i^\perp$ one is among the c_i and $c_i^\perp$ and one is among the b_j and $b_j^\perp$.

However one cannot say that a proof of DICOG-RS, i.e. a sequence of derivations yielding a dicograph $R[a_1, a_1^\perp, \ldots, a_n, a^n]$ with cuts (i.e. with a sub dicograph term $K \otimes K^\perp$) may be turned into a DICOG-RS derivation whose final dicographs is D restricted to the atoms that are neither in K not in $K^\perp$. Indeed the atoms in K and $K^\perp$ vanish during the process and none of the rewrite rules is able to do so — furthermore if one looks at step by step cut elimination, it precisely uses the prohibited rewriting rule $\otimes\parr 4$!

We shall see later that in fact cut elimination holds for proof nets that are dicographs of atoms but without any inductive notion of derivation.

4 Proof nets

This section presents proof structures and nets (the correct proof structures), in an abstract and algebraic manner, without links nor trip conditions: such proof strctures and nets are called handsome proof structures and nets. Basically proof nets consists in a dicograph R of atoms representing the conclusion formula, and axioms that are disjoint pairs of dual atoms constituting a partition B of the atoms of R. The proof net can be viewed as an edge bi-coloured graph: the dicograph is represented by R arcs and edges (Red and Regular in the pictures), while the axioms B (Blue and Bold in the picture). In such a setting, the correctness criterion expresses some kind of orthogonality between R and B. A proof net can also be viewed as a term, axioms being denoted by indices used exactly twice on dual atoms.

4.1 Handsome pomset proof nets

In fact, proof nets have (almost) been defined above!

A pomset logic *handsome proof structure* or DICOG-PN is a dicograph R over a (multi)set of $2n$ atoms, $\{a_1, \ldots, a_n, a_1^\perp, \ldots, a_n^\perp\}$, i.e. n propositional letters and their n duals. it is fairly possible that two atoms have the same name, i.e. it is a multiset of atoms. Let us call B the binary relation $\{(a_i, a_i^\perp)|1 \leq i \leq n\} \uplus \{a_i^\perp, a_i)|1 \leq i \leq n\}$ or simply $\{a_i\text{–}a_i^\perp\}$ using the notations of the previous section 2.2; observe that no two B edges are incident, and that each point is incident to exactly one edge in B: the B edges constitute a perfect matching of the whole graph with both B edges and R edges and arcs.

Given two proof structures π and π' whose atoms and axioms are the same, and whose conclusion formulas F and F' only differ because of the associativity

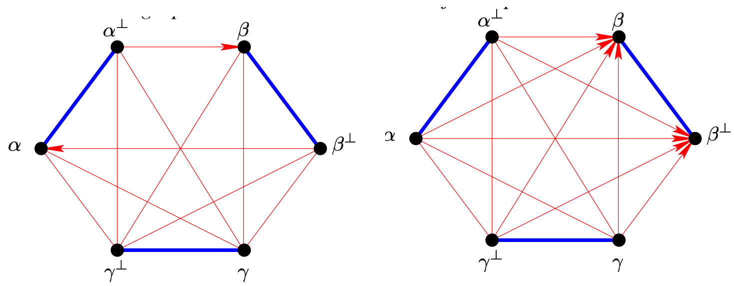


Figure 3: A proof structure containing a chordless alternate elementary (on the left) and a proof net without any chordless alternate elementary path (on the right).

of $⅋, <, \otimes$ and the commutativity of $⅋, \otimes$, the proof structures π and π' are equal — while in proof structures with links they would be different.

Correctness criterion 1 *A handsome proof structure is said to be a proof net whenever every elementary circuit (directed cycle) of alternating edges in R and in B contains a chord — an edge or arc connecting two points of the circuit but not itself nor its reverse in the circuit. In short, every* AE *circuit contains a chord. Observe that this chord cannot be in B, hence it is in R, and it can either be an R arc or an R edge.*

Theorem 5 (Nguyên) *Recently it was established that checking whether a proof structure satisfies the above correctness criterion is coNP complete [25].*

Theorem 6 *Given a proof net (B, R) if $R \rightsquigarrow R'$ (so $R' \subset R$) using rewriting rules of Figure 2.3 $\underline{\text{except } \otimes⅋4}$ then (B, R') is a proof net as well, i.e. all the rewrite rule preserve the correctness criterion on page 15.*

Proof. See [40, 41]. □

It is easily seen that in general $\otimes⅋4$ does not preserve correctness:

$B = \{a\text{–}a^\perp, b\text{–}b^\perp\}$ $R = \bar{B} = (a⅋a^\perp) \otimes (b⅋b^\perp) = \{a\text{–}b, a\text{–}b^\perp, a^\perp\text{–}b^\perp, a^\perp\text{–}b\}$. Using $\otimes⅋4$, $R = \bar{B}$ rewrites to
$R' = (a \otimes b)⅋(a^\perp \otimes b^\perp) = \{a\text{–}b, a^\perp\text{–}b^\perp\}$, and the proof net (B, R') contains the chordless æ circuit $(a, a^\perp) \in B, (a^\perp, b^\perp) \in R', (b^\perp, b) \in B, (b, a) \in R$.

Observe that it does *not* mean that every correct proof net (B, R) with axioms B can be obtained from $(B, \bar{B})$ by the allowed rewrite rules (all but $\otimes⅋4$) where $\bar{B}$ is $\widehat{\otimes}_i(a_i \widehat{⅋} a_i^\perp)$. Indeed, since $R \subset \bar{B}$ it is known that $\bar{B} \rightsquigarrow R$ but one cannot tell that $\otimes⅋4$ is not used. Indeed, as shown above $\otimes⅋4$ does not preserve correctness but it may happen:

As indicated in section 2.2 we write $a - b$ for the edge or par of opposite arcs $(a, b), (b, a)$

$B = \{a\text{–}a^\perp, b\text{–}b^\perp, c\text{–}c^\perp, d\text{–}d^\perp\}$

$R = (a^\perp \mathbin{\widehat{\otimes}} b^\perp) \mathbin{\widehat{\parr}} ((a \mathbin{\widehat{\parr}} b) \mathbin{\widehat{\otimes}} (c \mathbin{\widehat{\parr}} d)) \parr c^\perp \parr d^\perp = \{a^\perp\text{–}b^\perp, b\text{–}c, b\text{–}d, a\text{–}c, a\text{–}d\}$
$R' = (a^\perp \mathbin{\widehat{\otimes}} b^\perp) \mathbin{\widehat{\parr}} ((a \mathbin{\widehat{\otimes}} c) \mathbin{\widehat{\parr}} (b \mathbin{\widehat{\otimes}} d)) \parr c^\perp \parr d^\perp = \{a^\perp\text{–}b^\perp, b\text{–}c, b\text{–}d, a\text{–}c, a\text{–}d\}$
(B, R) is correct, and using $\otimes\parr 4$ it rewrites to (B, R') which is correct as well.

4.2 Cut and cut-elimination

What about the cut rule? This calculus has no rules in the standard sense, in particular no binary rules that would combine a K and a $K^\perp$. A cut is a tensor $K \otimes K^\perp$ which never is inserted inside a formula.

So a cut in this setting simply is a symmetric series composition $K \mathbin{\widehat{\otimes}} K^\perp$ in a dicograph whose form is $T \mathbin{\widehat{\parr}} (K \mathbin{\widehat{\otimes}} K^\perp)$. Assume the atoms of K are $\{a_1, \ldots, a_n\}$ so atoms of $K^\perp$ are $\{a_1^\perp, \ldots, a_n^\perp\}$. Cut-elimination consist in suppressing all edges and arcs between two atoms of K, all edges and arcs between two atoms of $K^\perp$, and all edges $a_i, a_j^\perp$ with $i \neq j$ — so the only edges incident to a_i are $a_i, a_i^\perp$ (call those edges atomic cuts) and $a_i x$ with x neither in K not in $K^\perp$. If, in this graph, an atom a is in the B relation with an $a^\perp$ in $K \cup K^\perp$, then the result of cut elimination is the closest point not in K nor in K' reached by an alternating sequence of B-edges and elementary cuts starting from a – observed that this point is necessarily named $a^\perp$, that we call its cut neighbour. To obtain the proof resulting from cut-elimination suppress all the atoms of K and $K^\perp$ as well as the incident arcs and edges and connect every atom to its cut neighbour with a B edge.

Theorem 7 *Cut elimination preserves the correctness criterion of* DICOG-PN *proof nets and consequently the f* DICOG-PN *proof nets enjoy cut-elimination.*

Proof. The preservation of the absence of chordless æ circuit during cut elimination is proved in [40, 41]. □

4.3 From sequent calculus and rewrite proofs to DICOG-PN

Proofs of the sequent calculus given in figure 2 are easy turn into a DICOG-PN proof net inductively. Such a derivation starts with axioms $\vdash a_i, a_i^\perp$ as it is well known, and in any kind of multiplicative linear logic the atoms a_i and $a_i^\perp$ that can be traced from the axiom that introduced them to the conclusion sequent, which, after some unfolding can be viewed as a dicograph of atoms R. The DICOG-PN proof structure corresponding to the sequent calculus proof simply is (B, R), and fortunately is a correct proof net.

Proposition 2 *A proof of sequent calculus corresponds to a* DICOG-PN *i.e. to a handsome proof structure without chordless alternate elementary path, i.e. into a handsome proof net.*

Proof. By induction on the proof, we showed in shown in [40, 41] that the neither the rules nor the the unfolding can introduce a chordless AE cycle. □

The above result also yield cut elimination for the sequent calculus. Indeed, proof nets obtained by cut-elimination from a proof net issued from the sequent calculus also are issued from the sequent calculus.

The derivation by dicograph rewriting DICOG-RS also only yield correct proof structures.

Proposition 3 *Any proof obtained by rewriting from AX_n yields a handsome proof structure without chordless alternate elementary path, i.e. into a* DICOG-*PN.*

Proof. Observe that AX_n satisfies the criterion, so because of theorem 6, the result is clear. □

5 Denotational semantics of pomset logic within coherence spaces

Denotational semantics or categorical interpretation of a logic is the interpretation of a logic in such a way that a proof d of $A \vdash B$ is interpreted as a morphism $[\![d]\!]$ from an object $[\![A]\!]$ to an object $[\![B]\!]$ in such a way that $[\![d]\!] = [\![d']\!]$ whenever d reduces to d' by (the transitive closure) of β-reduction or cut-elimination. A proof d of $\vdash B$ (when there is no A) is simply interpreted as a morphism from the terminal object $\mathbf{1}$ to B. More details can be found in [17, 8].

Once the interpretation of propositional variables is defined, the interpretation of complex formulas is defined by induction on the complexity of the formula. The set $Hom(A, B)$ of morphisms from A to B is in bijective correspondence with an object written $[\![A \multimap B]\!]$. Morphisms are defined by induction on the proofs and one has to check that the interpretations of a proof before and after one step of cut elimination is unchanged.

For intuitionistic logic, the category is cartesian closed, and for classical logic, at least simply, it is impossible[6]. Regarding linear logic, a categorical interpetation takes place in a monoidal closed category (with monads for the exponentials of linear logic).

5.1 Coherence spaces

The category of coherence spaces is a concrete category: objects are (countable) sets endowed with a binary relation, and morphimms are linear maps. It interprets the proofs up to cut-elimination or β reduction initially propositional intuitionistic logic and propositional linear logic (possibly quantified). Actually, coherence spaces are tightly related to linear logic: indeed, linear logic arose from this particular semantics, invented to model second order lambda calculus

[6]The fact that cartesian closed categories with involutive negation have at most one morphim between any two object is known as Joyal argument (see e.g. [17]); however there are complicated solutions like Selinger's control categories [47] for classical deductive systems that "control" the non determinism of classical cut elimination,like Parigot's $\lambda\mu$ calculus, [26].

i.e. quantified propositional intuitionistic logic [6]. Coherence spaces are themselves inspired from the categorical work on ordinals by Jean-Yves Girard; they are the binary qualitative domains.

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 $|A| \times |A|$ noted $\alpha \coh \alpha'[A]$ or simply $\alpha \coh \alpha'$ when A is clear.

The following notations are common and useful:

$$\alpha \frown \alpha'[A] \text{ iff } \alpha \coh \alpha'[A] \text{ and } \alpha \neq \alpha'$$
$$\alpha \asymp \alpha'[A] \text{ iff } \alpha \not\coh \alpha'[A] \text{ or } \alpha = \alpha'$$
$$\alpha \smile \alpha'[A] \text{ iff } \alpha \not\coh \alpha'[A] \text{ and } \alpha \neq \alpha'$$

A proof of A is to be interpreted by a *clique* of the corresponding coherence spaces A, a *cliques* being a set of pairwise coherent tokens in $|A|$ — we write $x \in A$ for $x \subset |A|$ and for all $\alpha, \alpha' \in x$ $\alpha \coh \alpha'$. Observe that forall $x \in A$, if $x' \subset x$ then $x' \in A$. A linear morphism F from A to B is a morphism mapping cliques of A to cliques of B such that:

- $\forall x \in A (x' \subset x) \Rightarrow F(x') \subset F(x)$
- Let $(x_i)_{i \in I}$ be a family of pairwise compatible cliques that is to say $\forall i, j \in I (x_i \cup x_j) \in A$ then $F(\cup_{i \in I} x_i) = \cup_{i \in I} F(x_i)$.[7]
- $\forall x, x' \in A$ if $(x \cup x') \in A$ then $F(x \cap x') = F(x) \cap F(x')$.

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

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

Those two rules are equivalent when contraction and weakening are allowed. The multiplicatives (contexts are split, $\otimes$ above) and the additives (contexts are duplicated, & above). Regarding denotational semantics, the web of the coherence space associated with a formula $A * B$ with $*$ a *multiplicative* connective is the Cartesian product $|A| \times |B|$ of the webs of A and B — while it is the disjoint union of the webs of A and B when $\bullet$ is additive.

Negation is a unary connective which is both multiplicative and additive: $|\neg A| = |A|$ and $\alpha \coh' \alpha[A^\perp]$ iff $\alpha \asymp \alpha'[A]$

One may wonder how many binary multiplicatives 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 ones that are covariant functors in both A and B — indeed there is a negation, hence a contravariant connective in A is a

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

covariant connective in $A^\perp$. Hence when both components are $\asymp$ so are the two couples, and when they are both coherent, so are the two couples.

To define a multiplicative connective, is to define when $(\alpha,\beta) \mathrel{\frown\mkern-16mu\smile} (\alpha',\beta')[A{*}B]$ in function of $\alpha \mathrel{\frown\mkern-16mu\smile} \alpha'[A]$ and $\beta \mathrel{\frown\mkern-16mu\smile} \beta'[B]$, so to fill a nine cell table — however if $*$ is assumed to be covariant in both its argument, seven out of the nine cells are filled.

$A * B$	$\smile$	$=$	$\frown$
$\smile$	$\smile$	$\smile$	NE?
$=$	$\smile$	$=$	$\frown$
$\frown$	SW?	$\frown$	$\frown$

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

$A \parr B$	$\smile$	$=$	$\frown$
$\smile$	$\smile$	$\smile$	$\frown$
$=$	$\smile$	$=$	$\frown$
$\frown$	$\frown$	$\frown$	$\frown$

and

$A \otimes B$	$\smile$	$=$	$\frown$
$\smile$	$\smile$	$\smile$	$\smile$
$=$	$\smile$	$=$	$\frown$
$\frown$	$\smile$	$\frown$	$\frown$

However if we don't ask for the connective to be commutative we have a third connective $A < B$ (and actually a fourth connective $A > B$ which is $B < A$)

$A < B$	$\smile$	$=$	$\frown$
$\smile$	$\smile$	$\smile$	$\smile$
$=$	$\smile$	$=$	$\frown$
$\frown$	$\frown$	$\frown$	$\frown$

and

$A > B$	$\smile$	$=$	$\frown$
$\smile$	$\smile$	$\smile$	$\frown$
$=$	$\smile$	$=$	$\frown$
$\frown$	$\smile$	$\frown$	$\frown$

This connective generalises to partial orders. Assume we have an SP order $T[A_1,\ldots,A_n]$ on the formulas $A_1,\ldots,A_n$ — T can be defined with $\parr$ and $<$ — two tuples $(\alpha_1,\ldots,\alpha_n)$ and $(\alpha'_1,\ldots,\alpha'_n)$ of the web $|T[A_1,\ldots,A_n]|$ are strictly coherent whenever $\exists i\ \big(\alpha_i \frown \alpha'_i\ \&\ \big(\forall j > i \alpha_j \frown \alpha'_j\big)\big)$.

Linear implication, which can be defined as $A^\perp \parr B$ is :

$A \multimap B$	$\smile$	$=$	$\frown$
$\smile$	$\frown$	$\frown$	$\frown$
$=$	$\smile$	$=$	$\frown$
$\frown$	$\smile$	$\smile$	$\frown$

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

One can observe that $\{((\alpha,(\beta,\gamma)),((\alpha,\beta),\gamma)) \mid \alpha \in |A|, \beta \in |B|, \gamma \in |C|\}$ defines a linear isomorphism from $A{<}(B{<}C))$ to $(A{<}B){<}C$, that $\{((\alpha,\beta),(\alpha,\beta)) \mid \alpha \in |A|, \beta \in |B|\}$ defines a linear morphims from $A \otimes B$ to $A < B$ and the same set of pairs of tokens also defines a linear morphims from $A < B$ to $A \parr B$. However, for general coherence spaces A and B there is no canonical linear map from $A < B$ to $B < A$.

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 model in the multiplicative case (without before), see [22].

The before connective is issued from coherence semantics, hence it is a good idea to explore the coherence semantics of the logical calculi we designed for pomset logic, to see whether they are sound.

5.2 A sound and faithful interpretation of proof nets in coherence spaces

An important criterion comforting the design of the deductive systems for pomset logic is that those systems are sound w.r.t. coherence semantics — in addition to cut-elimination discussed previously. We shall here interprets a proof net with conclusion T (a formula or a dicograph of atoms) as a clique of the corresponding coherence space T.

Computing the semantics of a cut-free proof net is rather easy, using Girard's experiments but from axioms to conclusions as done in [33, 38].

However, we define the interpretation of a proof structure (non necessarily a proof net) as a set of tokens of the web of the conclusion formula. Assume the proof structure is $B = \{a_i - a_i^\perp | 1 \leq i \leq n\}$and that each of the a_i as a corresponding coherence space a also denoted by a_i. For each a_i choose a token $\alpha_i \in |a_i|$. If the conclusion is a dicograph T replacing each occurrence of a_i and each occurrence of $a_i^\perp$ with α_i yields a term, which when converting $x * y$ (with $*$ being one of the connectives, $\parr, <, \otimes$) 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 $[\![\pi]\!]$ of the normal proof structure π is the set of all the results of the experiments on π. One has the following result that Lambek appreciated, because it replaces graph theoretical considerations with algebraic properties:

Theorem 8 *A proof structure π with conclusion T is a proof net (contains no chordless* AE*-circuit) and only if its interpretation $[\![\pi]\!]$ is a clique of the coherence space T (is a semantic object).*

Proof. The proof is a consequence of:

- both folding and unfolding (see subsection 7.1 or [39, 40, 41]) preserve correctness
- semantic characterisation of proof nets with links correctness is proved in [33] for MLL and pomset logic — the published version left out pomset logic [38].

□

The actual result we proved is a bit more: in order to check correctness one only has to use a given four-token coherence space, and this provide a way to check correctness, which is oh an exponential complexity in accordance with the recent results by Nguyên [25].

When π is not normal, i.e. includes cuts, not all experiments succeed and provide results: an experiment is said to succeed when in every cut $K cut K^{\perp}$ the value α on a in 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 $[\![\pi]\!]$ of the normal proof net π. Whenever π reduces to π' by cut elimination $[\![\pi]\!] = [\![\pi']\!]$. That way one is able to predict that a proof structure will reduce to a proof net or not,[8] without actually performing cut elimination:

Theorem 9 *Let π be a proof structure and let π^* be its normal form; then π^* is a proof net plus zero or more loops (cut between two atoms that are connected with an axiom) whenever two succeeding experiments of π have coherent or equal results.*

Proof. See [33, 38]. □

6 Sequentialisation with pomset sequents or dicographs sequents

In 2001, Lambek noticed the absence of sequent calculus in my habilitation [42]. Although there is one in my PhD that was refined later to only use SP orders, I did not put it forward because the proof net calculus enjoys much more mathematical properties and is richer in the sense that it does not covers all the proof nets. I tried and Sylvain Pogodalla and Lutz Straßburger as well, to prove that every correct proof net is the image of a proof in the sequent calculus — the one given here or some variant.

The SP-pomset calculus of sequents sequent calculus presented in Figure 2 is clearly equivalent to the dicograph sequent calculus with dicographs of atoms as sequents; in the dicograph sequent calculus, the symmetric series compositons $\widehat{\otimes}$ may well be used on contexts, as the $\widehat{\wp}$ and $\widehat{<}$ rule, and all connective introduction rules consists in internalising the $\widehat{*}$ operation inside a formula as a $*$ connective. This calculus is shown in figure 4. Observe that entropy does not allow inclusion of dicograph in general, but only of an outer SP-order; indeed, in general, dicograph inclusion does not preserve correctness, as explained in subsection 6.

An induction on either sequent calculus given in this paper shows that:

[8] Proof nets reduces to proof nets,, occrectness is preserved under cut-elimination, but an incorrect proof structure may well reduce to a proof net.

$$\frac{}{\vdash a \widehat{\wp} a^{\perp}}\text{ axiom}$$

$$\frac{\vdash \Gamma \qquad \vdash \Delta}{\vdash \Gamma \mathbin{\widehat{<}} \Delta}\ dimix$$

$$\frac{\vdash O[\Gamma_1, \ldots, \Gamma_p]}{\vdash O'[\Gamma_1, \ldots, \Gamma_p]}\text{ entropy}\begin{cases}\text{with } \Gamma_i\text{:dicographs,} \\ O, O' \text{ SP-orders, } O' \subset O\end{cases}$$

$$\frac{\vdash A\widehat{\wp}\Gamma \qquad \vdash B\widehat{\wp}\Delta}{\vdash \Gamma\widehat{\wp}(A \mathbin{\widehat{\otimes}} B)\widehat{\wp}\Delta} \otimes \text{ / cut when } A = B^{\perp}$$

$$\frac{\vdash \Gamma[A\widehat{\wp}B]}{\vdash \Gamma[A\wp B]} \wp \text{ if } A \ddot{\sim} B \qquad \frac{\vdash \Gamma[A \mathbin{\widehat{<}} B]}{\vdash \Gamma[A < B]} < \text{ if } A \overset{\rightarrow}{\sim} B \qquad \frac{\vdash \Gamma[A \mathbin{\widehat{\otimes}} B]}{\vdash \Gamma[A \otimes B]} \wp \text{ if } A \bar{\sim} B$$

Figure 4: Dicograph sequent calculus with dicographs of atoms as sequents

Proposition 4 *Let δ be a proof a dicograph sequent R, and let $\pi_\delta = (B, R)$ be the corresponding proof net. Then the axioms and atoms of π_δ can be partitioned into two classes $\Pi_1 = (a_i - a_i^{\perp})_{i \in I_1}$ and $\Pi_2 = (a_i - a_i^{\perp})_{i \in I_2}$ in such a way that either:*

1. *there are only arcs from Π_1 to Π_2*
2. *the only edges between Π_1 and Π_2 are a $\widehat{\otimes}$ connection: calling $R_1 = R\restriction_{\Pi_1}$ and $R_2 = R\restriction_{\Pi_2}$, $R_1 = A_1\widehat{\wp}T_1$, $R_1 = A_2\widehat{\wp}T_2$, and $R = (A_1 \mathbin{\widehat{\otimes}} A_2)\widehat{\wp}T_1\widehat{\wp}T_2$*

Proposition 5 *There does exist a proof net without any sequent calculus proof for example the one in figure 5.*[9]

Proof. First one as to observe that the proof structure in figure 5 is a proof net, i.e. contains no chordless alternate elementary circuit: indeed, it contains no alternate elementary circuit.

Because of proposition 4, there should exists a partition into two parts with

1. either only arcs from one part to the other part,
2. or a tensor connection between the two parts.

[9] If i remember well, this correct proof net or dicograph is derivable from $(a\widehat{\wp}a^{\perp})\widehat{\otimes}(b\widehat{\wp}b^{\perp})\widehat{\otimes}(c\widehat{\wp}c^{\perp}) \mathbin{\widehat{\otimes}} (d\widehat{\wp}d^{\perp}) \mathbin{\widehat{\otimes}} (e\widehat{\wp}e^{\perp}) \mathbin{\widehat{\otimes}} (f\widehat{\wp}f^{\perp})$ by means of the rewriting rules that preserves the correctness that are all rules of 2.3 but $\otimes\wp 4$. The derivation will be given in an ulterior version of the paper.

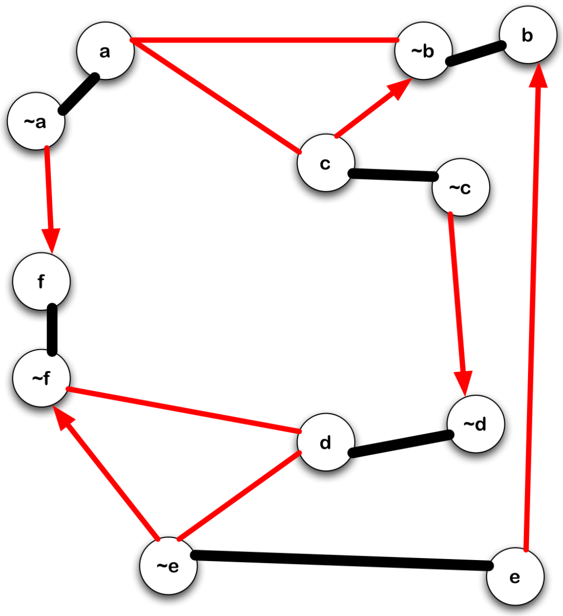


Figure 5: A proof net with no corresponding sequent calculus proof (found with Lutz Straßburger)

If the first case applies, i.e. if there were a partition into two parts with only arcs from one to another, all by vertices connected with an undirected edge, be it a B or an R edge, should be in the same component. $a, a^\perp, b, b^\perp, c, c, c^\perp$ should be in the same component say Π_1 and $f, f^\perp, d, d^\perp, e, e^\perp$ should be in the same component say Π_2, but this is impossible because there are both an R arc from Π_1 to Π_2, e.g. $a^\perp \to f$, and en R arc from Π_2 to Π_1, e.g. $e \to b$. So the first case does not apply.

Because the first case does not apply, there should exist two parts, with a tensor rule as the only connection between two parts. The two possible tensors are $a \mathbin{\widehat{\otimes}} (c \mathbin{\widehat{<}} b^\perp)$ and $d \mathbin{\widehat{\otimes}} (e^\perp \mathbin{\widehat{<}} f^\perp)$, but it is impossible:

- $a \mathbin{\widehat{\otimes}} (c \mathbin{\widehat{<}} b^\perp)$ cannnot be the only connection between the two parts, as there exists an undirected path fro c to a not using any of the two tensor R edges:
$$c \xleftrightarrow{B} c^\perp \xrightarrow{R} d^\perp \xleftrightarrow{B} d \xleftrightarrow{B} f^\perp \xleftrightarrow{R} f \xleftarrow{R} a^\perp$$
- $d \mathbin{\widehat{\otimes}} (e^\perp \mathbin{\widehat{<}} f^\perp)$ cannnot be the only connection between the two parts, as there exists an undirected path from $f^\perp$ to d not using any of the two tensor R edges:
$$f^\perp \xleftrightarrow{B} f \xleftarrow{R} a^\perp \xleftrightarrow{B} a \xleftrightarrow{R} c \xleftrightarrow{B} c^\perp \xrightarrow{R} d^\perp \xleftrightarrow{B} d$$

$\square$

Question 1 *We may wonder whether all proof nets, including the one in figure 5, can be obtained from $AX_n = \widehat{\otimes}_{i\in I}(a_i \widehat{\parr} a_i^\perp)$ using only the correct rewriting rules (inclusion patterns) of 2.3 (all of them but $\otimes\parr 4$). This question is equivalent to another question, namely the equivalence between pomset logic as defined by* DICOG-PN *with the BV calculus of Guglielmi and Straßburger [11].*

7 Grammatical use

Relations like dicographs have pleasant algebraic properties but when it comes to combining trees as in grammatical derivations, it is better to view the trees in order to have some intuition. So we first present proof nets with links before defining a grammatical formalism.

7.1 Proof nets with links

In order to define a grammar of pomset proof nets, if is easier to use proof nets with links which look like standard proof nets: the formula trees of the conclusions $T_1, \ldots, T_n$ with binary connectives $(\parr, \otimes, <)$ and axioms linking dual atoms, together with an SP partial order on the conclusions $T_1, \ldots, T_n$.

It is quite easy to turn a DICOG-PN proof net into a pomset proof net using folding of subsection 2.4 — and vice-versa using unfolding. A dicograph proof net $\pi = (B, R)$ with R being $S[T_1, \ldots, T_n]$ with S containing no $\widehat{\otimes}$ symbol — S is an SP order — corresponds to a pomset proof net π^{SP} with conclusions $T_1^f, \ldots, T_n^f$ where T_i^f is the formula corresponding to T_i obtained by replacing an operation on dicograph $\widehat{*}$ with the corresponding multiplicative connective $*$ — $*$ being one of the connective $\otimes, <, \parr$. There usually are many ways to write a dicograph R as a term $S[T_1, \ldots, T_n]$ depending on the associativity of $\otimes, \parr, <$, commutativity of $\otimes, \parr$ and the n may vary when the outer most $\widehat{\parr}$ and $\widehat{<}$ are turned into $\parr$ and $<$ connectives or not (as it is the case for $\parr$ in usual proof nets for MLL). In case the outer most connective of R is $\widehat{\otimes}$, π necessarily has a single conclusion, $R = T_1$, and S is the trivial SP order on one formula.

The transformation from π to π^{SP} can be done "little by little" by allowing "intermediate" proof structures whose conclusion is a dicograph of *formulas*. Such a proof structure is said to be correct whenever every æ circuit contains a chord, the formula trees being bicoloured as in figure 6 — in figure 7 π_1 is the DICOG-PN proof net, while π_4 is a pomset proof net with links having a single conclusion.

Let $\pi = (B, D[F_1, \ldots, F_p])$ with D a dicograph on the formulas $F_1, \ldots, F_n$ be an intermediate proof structure . A folding of π is a simply a folding of $D[F_1, \ldots, F_p]$ as defined in subsection 2.4 (two equivalent formulas $F_i \widehat{*} F_j$ are replaced in D by one formula $F_i * F_j$). An unfolding of π is simply an unfolding of $D[F_1, \ldots, F_p]$ as defined in subsection 2.4 (a formula $F_i * F_j$ is replaced by two equivalent formulas $F_i \widehat{*} F_j$).

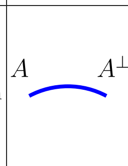
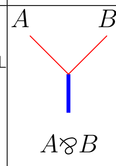
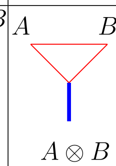
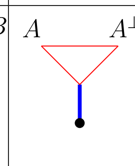
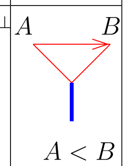
Name	axiom	$\parr$	$\otimes$	`Cut`	$<$
Premises	none	A and B	A and B	A and $A^\perp$	A and B
R&B-graph					
Conclusions	A and $A^\perp$	$A \parr B$	$A \otimes B$	none	$A < B$

Figure 6: RnB links

Proposition 6 *Let π_f and π_u be two intermediate proof structures, with π_u being an unfolding of π_f — or π_f being a folding of π_u. The two following properties are equivalent:*

- *π_u is correct.*
- *π_f is correct.*

Proof. This proof consists in a thorough examination of new AE circuits that may appear during the transformation and of the edges that are chords and that may vanish during the transformation. [41]. □

Because of the shape of the links the criterion "every æ circuit contains a chord" is easier to formulate for pomset proof nets–the relation between conclusion formulas is an SP order: "there is no AE circuit".

7.2 Gammars with partial proof nets

Alain Lecomte was aiming at extensions of the Lambek grammars that would handle relatively free word orders, discontinuous constituents and other tricky linguistics phenomena, but still within a logical framework — as opposed to CCG which extends AB grammars with *ad hoc* rewriting rules whose logical content is unclear. Grammars defined within a logical framework have at least two advantages: rules remain general and the connection with semantics, logical formulas and lambda terms is a priori more transparent. Following a suggestion by Jean-Yves Girard, Alain Lecomte contacted me just after I passed my PhD on pomset logic, so we proposed a kind of grammar with pomset logic. We explored such a possiblity in [18, 21, 19, 20] and it was later improved by Sylvain Pogodalla in [29] (see also [42]).

We followed two guidelines:

- Words are associated not with formulas but with partial proof nets with a tree-like structure, in particular they have a single output;
- word order is a partial order, an SP order described by the occurrences of the $<$ connective in the proof net.

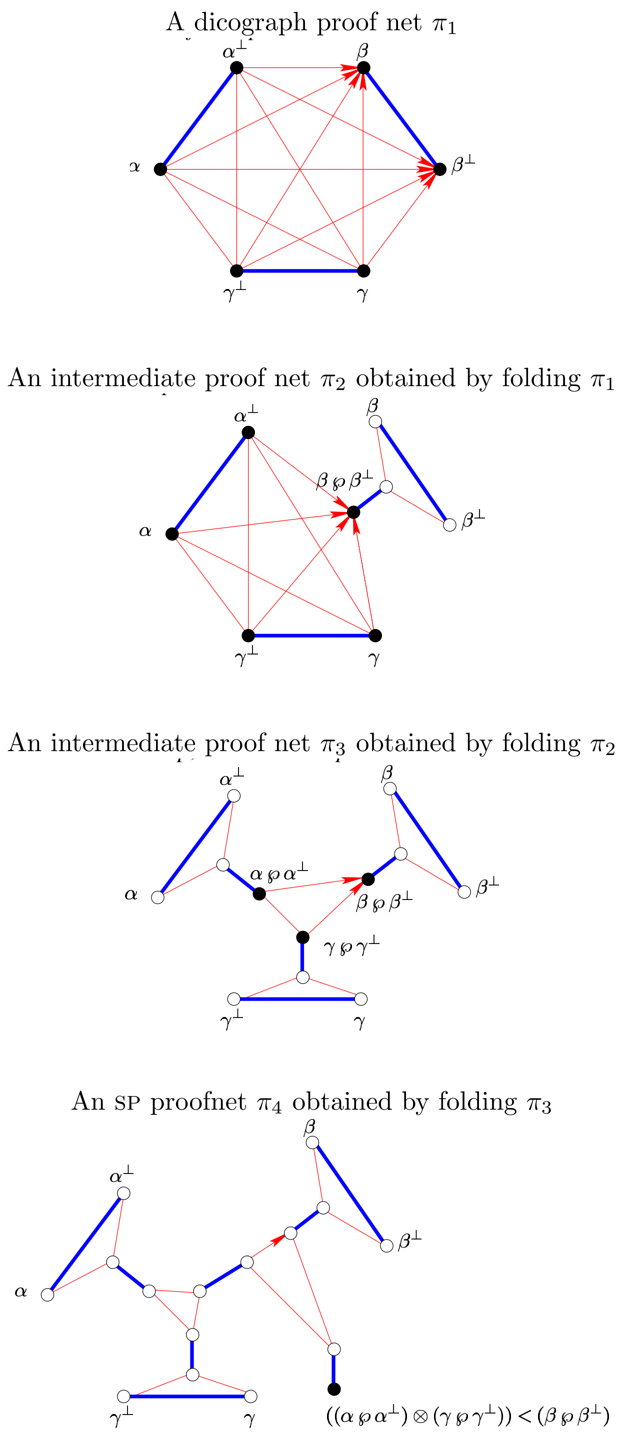


Figure 7: Progressively turning a dicograph proof net π_1 into a pomset proof net π_4 with one conclusion via some intermediate proof nets π_2 and π_3. The conclusions of the pomset proof net and of the intermediate proof nets are emphasised by filled black dots.

An analysis or *parse structure* is a combination of the partial proof nets into a complete proof net with output S. The two ways to combine partial proof nets are by "plugging" an hypothesis to the conclusion of another partial proof net, and to perform cuts between partial proof nets.

Given that words label axioms, instead of having a single B edge from a–$a^\perp$ we write a sequence of three edges, a B edge, an R edge, a B edge, the middle one being labelled with the word $a \underset{B}{\text{———}} \circ \frac{word}{R} \circ \underset{B}{\text{———}} a^\perp$; this little variant changes nothing regarding the correctness of the proof net in terms of æ paths.

Rather than lengthy explanations, let us give two examples of a grammatical derivation in this framework. One may notice in the examples that the partial pomset proof nets that we use in the lexicon are of a restricted form:

- there are just two conclusions:
  - the output b which is the syntactic category of the resulting phrase once the required "arguments" have been provided;
  - a conclusion $a^\perp \parr (X_1 \otimes Y_1) \parr \cdots \parr (X_n \otimes Y_n)$ without any $\otimes$ connective in the X_i;
- an axiom connects $a^\perp$ in the conclusion with an a in one of the X_i — with the corresponding word is the label of a;

In a first version we defined from the proof net an order between atoms (hence words) by "there exists a directed path" from a to b. However it is more convenient, in particular from a computational point of view, to label the proof net with SP orders of words. Doing so is a computational improvement but those labels are fully determined by the proof net, they contain no additional information. Here are the labelling rules:

- Initialisation:
  - $a^\perp$ is labelled with the one point SP order consisting of the corresponding word;
  - $X_i \otimes Y_i$ is labelled with an empty SP order.
- Propagation:
  - The two conclusions of a given axiom have the same label;
  - One of the two premisses of a tensor link is labelled with the SP order $R \widehat{\parr} S$ the other by R and the conclusion by S;
  - The conclusion of a $\parr$ link is labelled $R \widehat{\parr} S$ when the two premises are labelled R and S;
  - The conclusion of a $<$ link is labelled $R \mathrel{\widehat{<}} S$ when the two premises are labelled R and S;

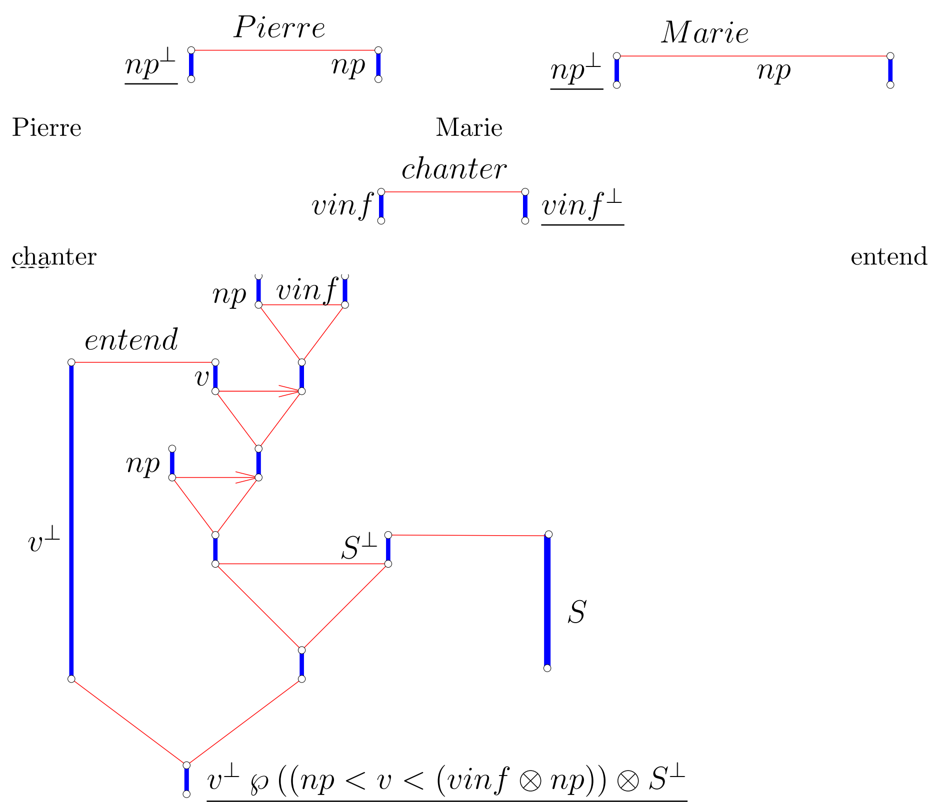
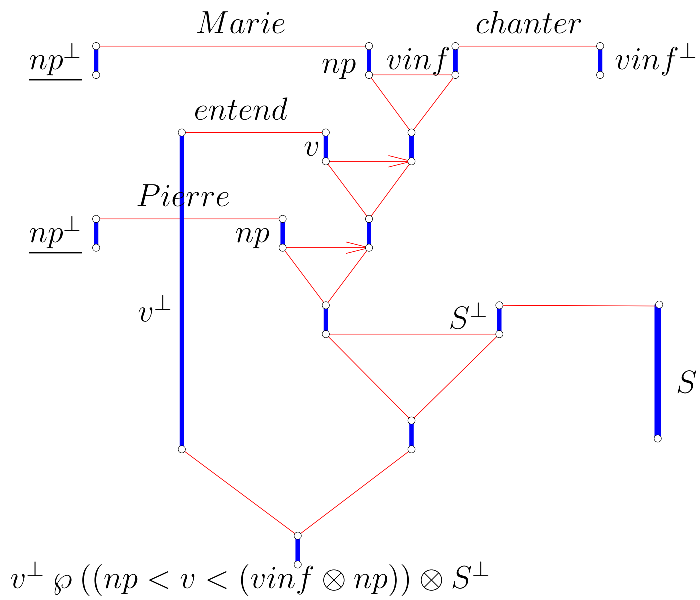


Figure 8: A lexicon with partial pomset proof nets



Pierre

Figure 9: Analysis of a relatively free word order sentence — order *Pierre* < *entend* < ($Marie \wp chanter$)

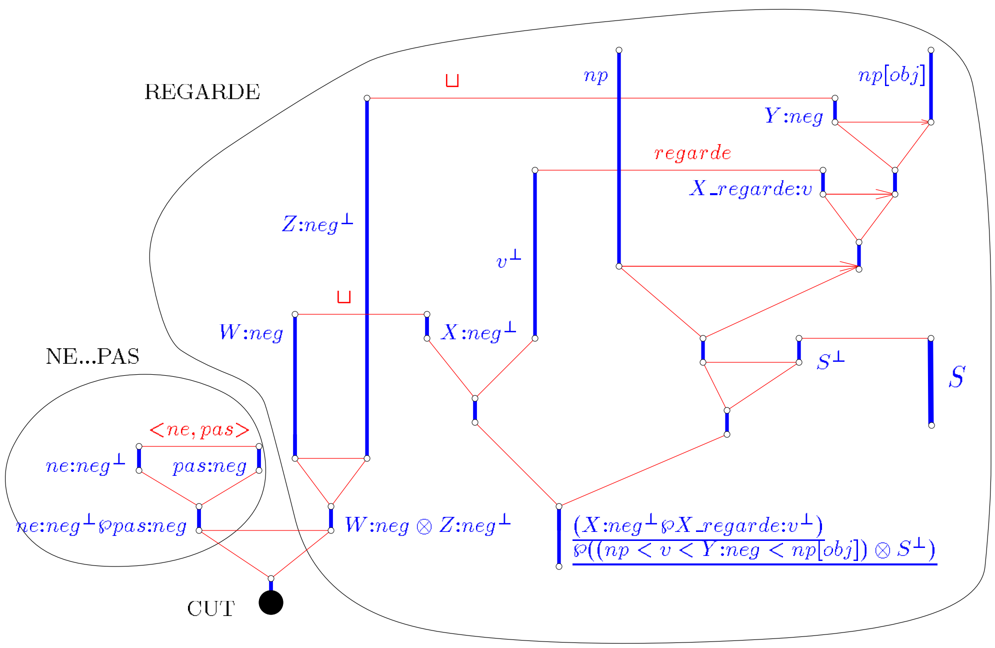
The propagation rules always succeed because of the correctness criterion and of the tree like structure of the partial proof nets. The propagation rules yield a complete labelling of the proof net and the SP order that labels the output S is the partial order over words.

We give an example of a lexicon of an analysis of a relatively free word order phenomenon in French — the lexicon is in figure 8 and the analysis in 9. One can say both "Pierre entend Marie chanter" (Pierre hears Mary singing) and "Pierre entend chanter Marie" (Pierre hears singing Mary). Indeed when there is no object French accepts that the subject is after the verb, e.g. in relatives introduced by the relative pronom "que/whom": "Pierre que regarde Marie chante." (Pierre that Mary watches sings" and "Pierre que Marie regarde chante." (Pierre that Marie watches sings). Observe that there is a single analysis for the different possible word orders and not a different analysis for each word order.

Using cuts, one is able, in addition to free word order phenomena, to provide an account of discontinuous constituents, e.g. French negation "ne . . . pas". During cut elimination, the label splits into two parts so "ne" and "pas" go to their proper places, as shown in figure 10.

It is difficult to say something on the generative capacity of this grammatical formalism, because it produces (or recognises) SP order of words and not chains of words — and there are not so many such grammatical formalisms, en

The proof net made from the partial proof nets NE...PAS (discontinuous constituent) and from the partial proof net REGARDE, before cut-elimination.



The proof net analysing NE REGARDE PAS, after reduction, the three words are in the proper order:

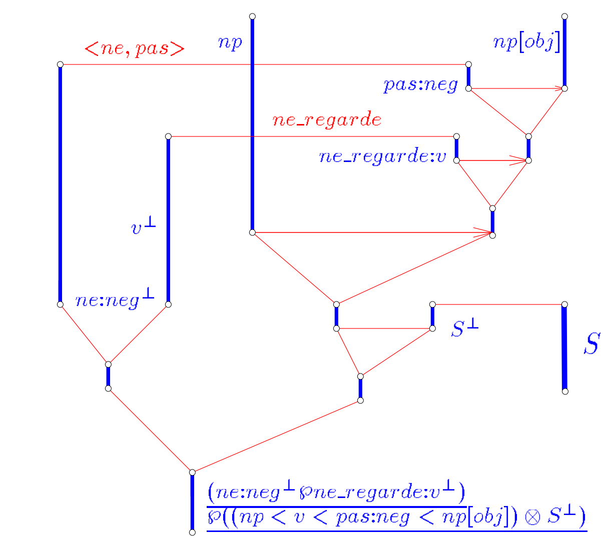


Figure 10: Handling discontinuous constituents in pomset proof nets

exception being [23].

Theorem 10 (Pogodalla) *Pomset grammars with a restricted form for partial pomset proof nets yielding trees and total word orders is equivalent to Lexicalised Tree Adjoining Grammars. [29]*

This is much more than languages that can be generated by Lambek grammars, that are context free. In both cases, parsing as proof search is NP complete – trying all the possibilities in pomset grammar is in NP (and likely to be NP complete), and provability Lambek calculus has been shown to be NP complete [28] — of course if the Lambek grammar is converted into an extremely large context free grammar using the result of Matti Pentus [27] parsing of Lambek grammars is polynomial, cubic or better in the number of words in the sentence.

Especially when using cuts and tree-like partial proof net this calculus is close to several coding of LTAG in non commutative linear logic à la Lambek-Abrusci [2].[10]

8 Conclusion and perspective

We presented an overview of pomset logic with both published and unpublished results. Pomset logic is a variant of linear logic, as the Lambek calculus is, and it can be used for modelling grammar, in particular for natural language as the Lambek calculus can.

Apart from this, as said in the introduction, Lambek calculus and pomset logic, are quite different, although they are both non commutative variants of (multiplicative) linear logic.

But perhaps the resemblance is more abstract than that. Indeed Lambek was surprised that with proof nets people intend to replace a syntactic calculus, an algebraic structure, with graphical or geometrical objects. However for pomset logic, the best presentation is certainly the calculus of dicographs, which are terms, and therefore belong to algebra. It is not surprising that Lambek preferred my algebraic correctness criterion [33, 38, 42] theorem (here theorem 8) with coherence spaces to the double trip condition of Girard of citeGir87.

This presentation is by no means the necrology of pomset logic. Indeed, Sergey Slavnov recently proposed a sequent calculus which is complete w.r.t. pomset proof nets.[48] In his sequent calculus, multisets of formulas are endowed with binary relations on sequences of n conclusions, and $<$ is a collapse of two connectives namely a $<$ that looks like $\otimes$ and a $<$ which looks like a $\parr$.

Lutz Straßburger who contributed to pomset logic with the counterexample, but also by looking at the similarity with the later born Deep Inference, also has new ideas in connection to software safety.

This gives excellent reasons to explore pomset logic again.

[10]The related work [13] which also encodes TAGs in non commutative linear logic à la Lambek-Abrusci, presented with natural deduction, requires *ad hoc* extensions of the non commutative linear logic like some crossing of the axioms which are excluded from those Lambek-Abruci logics [45, 34].

References

[1] Abrusci, V.M.: Phase semantics and sequent calculus for pure non-commutative classical linear logic. Journal of Symbolic Logic **56**(4), 1403–1451 (1991)

[2] Abrusci, V.M., Fouqueré, C., Vauzeilles, J.: Tree adjoining grammar and non-commutative linear logic. In: Retoré [36], pp. 13–17

[3] Abrusci, V.M., Ruet, P.: Non-commutative logic I: the multiplicative fragment. Annals of Pure and Applied Logic (1999)

[4] Bechet, D., de Groote, P., Retoré, C.: A complete axiomatisation of the inclusion of series-parallel partial orders. In: H. Comon (ed.) Rewriting Techniques and Applications, RTA'97, *LNCS*, vol. 1232, pp. 230–240. Springer Verlag (1997)

[5] Fleury, A., Retoré, C.: The mix rule. Mathematical Structures in Computer Science **4**(2), 273–285 (1994)

[6] Girard, J.Y.: The system F of variable types: Fifteen years later. Theoretical Computer Science **45**, 159–192 (1986)

[7] Girard, J.Y.: Linear logic. Theoretical Computer Science **50**(1), 1–102 (1987)

[8] Girard, J.Y., Lafont, Y., Taylor, P.: Proofs and Types. No. 7 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press (1988)

[9] Guglielmi, A.: Concurrency and plan generation in a logic programming language with a sequential operator. In: P. van Hentenryck (ed.) International Conference on Logic Programming, pp. 240–254. M.I.T. Press, Genova (1994)

[10] Guglielmi, A.: A calculus of order and interaction. Tech. Rep. WV-99-04, Dresden University of Technology (1999)

[11] Guglielmi, A., Straßburger, L.: Non-commutativity and MELL in the calculus of structures. In: L. Fribourg (ed.) CSL 2001, *Lecture Notes in Computer Science*, vol. 2142, pp. 54–68. Springer-Verlag (2001)

[12] Guglielmi, A., Straßburger, L.: A system of interaction and structure iv: The exponentials and decomposition. ACM transaction of computational logic **12**(4), 23 (2011)

[13] Joshi, A., Kulick, S.: Partial proof trees, resource sensitive logics and syntactic constraints. In: Retoré [36], pp. 21–42

[14] Kelly, D.: Comparability graphs. In: Rival [44]. Pp 3–40

[15] Lamarche, F., Retoré, C.: Proof nets for the Lambek calculus – an overview. In: V.M. Abrusci, C. Casadio (eds.) Third Roma Workshop: Proofs and Linguistics Categories – Applications of Logic to the analysis and implementation of Natural Language, pp. 241–262. CLUEB, Bologna (1996)

[16] Lambek, J.: The mathematics of sentence structure. American mathematical monthly pp. 154–170 (1958)

[17] Lambek, J., Scott, P.J.: Introduction to Higher Order Categorical Logic, *Cambridge Studies in Advanced Mathematics*, vol. 7. Cambridge University Press (1986)

[18] Lecomte, A., Retoré, C.: Pomset logic as an alternative categorial grammar. In: G. Morrill, R. Oehrle (eds.) Formal Grammar, pp. 181–196. FoLLI, Barcelona (1995)

[19] Lecomte, A., Retoré, C.: Words as modules and modules as partial proof nets in a lexicalised grammar. In: V.M. Abrusci, C. Casadio (eds.) Third Roma Workshop: Proofs and Linguistics Categories – Applications of Logic to the analysis and implementation of Natural Language, pp. 187–198. Bologna:CLUEB (1996)

[20] Lecomte, A., Retoré, C.: Logique des ressources et réseaux syntaxiques. In: D. Genthial (ed.) *4 ème conférence sur le Traitement automatique du langage naturel, TALN'97, pp. 70–83. Grenoble (1997)*

[21] Lecomte, A., Retoré, C.: Words as modules: a lexicalised grammar in the framework of linear logic proof nets. In: C. Martin-Vide (ed.) Mathematical and Computational Analysis of Natural Language — selected papers from ICML'96, Studies in Functional and Structural Linguistics*, vol. 45, pp. 129–144. John Benjamins publishing company (1998)*

[22] Loader, R.: Linear logic, totality and full completeness. In: LICS: IEEE Symposium on Logic in Computer Science (1994)

[23] Lodaya, K., Weil, P.: Series-parallel languages and the bounded-width property. Theor. Comput. Sci. **237***(1-2), 347–380 (2000). DOI 10.1016/S0304-3975(00)00031-1. URL* `https://doi.org/10.1016/S0304-3975(00)00031-1`

[24] Möhring, R.H.: Algorithmic aspects of comparability graphs. In: Rival [44], pp. 41–101

[25] Nguyên, L.T.D.: Proof nets through the lens of graph theory: a compilation of remarks (2019). URL `https://arxiv.org/abs/1912.10606`

[26] Parigot, M.: $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction proofs. In: S. Verlag (ed.) LPAR'92, LNCS*, vol. 624, pp. 190–201 (1992)*

[27] *Pentus, M.: Lambek grammars are context-free. In: Logic in Computer Science. IEEE Computer Society Press (1993)*

[28] *Pentus, M.: Lambek calculus is NP-complete. Tech. Rep. TR-2003005, CUNY - City University of New York (2003). Http://www.cs.gc.cuny.edu/tr/*

[29] *Pogodalla, S.: Lexicalized proof-nets and TAGs. In: M. Moortgat (ed.) Logical Aspects of Computational Linguistics, LACL'98, selected papers, no. 2014 in LNCS/LNAI, pp. 230–250. Springer-Verlag (2001)*

[30] *Pogodalla, S., Retoré, C.: Handsome non-commutative proof-nets: perfect matchings, series-parallel orders and hamiltonian circuits. In: Categorial grammars – an efficient tool for natural language processing (2004). URL* `http://hal.inria.fr/inria-00071248`

[31] *Retoré, C.: Graph theory from linear logic: aggregates. Prépublication 47, Equipe de Logique, Université Paris 7 (1993)*

[32] *Retoré, C.: Réseaux et séquents ordonnés. Thèse de Doctorat, spécialité Mathématiques, Université Paris 7 (1993). URL* `https://tel.archives-ouvertes.fr/tel-00585634v1`

[33] *Retoré, C.: On the relation between coherence semantics and multiplicative proof nets. Rapport de Recherche RR-2430, INRIA (1994). URL* `https://hal.inria.fr/inria-00074245`

[34] *Retoré, C.: Calcul de Lambek et logique linéaire. Traitement Automatique des Langues* ***37****(2), 39–70 (1996)*

[35] *Retoré, C.: Perfect matchings and series-parallel graphs: multiplicative proof nets as R&B-graphs. In: J.Y. Girard, M. Okada, A. Scedrov (eds.) Linear'96,* Electronic Notes in Theoretical Science*, vol. 3. Elsevier (1996). URL* `https://doi.org/10.1016/S1571-0661(05)80416-5`*. Http://www.elsevier.nl/*

[36] *Retoré, C. (ed.): Logical Aspects of Computational Linguistics, LACL'96,* LNCS/LNAI*, vol. 1328. Springer-Verlag (1997)*

[37] *Retoré, C.: Pomset logic: a non-commutative extension of classical linear logic. In: P. de Groote, J.R. Hindley (eds.) Typed Lambda Calculus and Applications, TLCA'97,* LNCS*, vol. 1210, pp. 300–318 (1997)*

[38] *Retoré, C.: A semantic characterisation of the correctness of a proof net. Mathematical Structures in Computer Science* ***7****(5), 445–452 (1997)*

[39] *Retoré, C.: Handsome proof-nets: R&B-graphs, perfect matchings and series-parallel graphs. Rapport de Recherche RR-3652, INRIA (1999). URL* `https://hal.inria.fr/inria-00073020`

[40] Retoré, C.: Pomset logic as a calculus of directed cographs. In: V.M. Abrusci, C. Casadio (eds.) Dynamic Perspectives in Logic and Linguistics: Proof Theoretical Dimensions of Communication Processes,Proceedings of the 4th Roma Workshop, pp. 221–247. Bulzoni, Roma (1999). See also [41]

[41] Retoré, C.: Pomset logic as a calculus of directed cographs. Rapport de Recherche RR-3714, INRIA (1999). URL https://hal.inria.fr/inria-00072953. See also [40]

[42] Retoré, C.: Logique linéaire et syntaxe des langues. Mémoire d'Habilitation à Diriger des Recherches, Université de Nantes (2002). https://tel.archives-ouvertes.fr/tel-00354041

[43] Retoré, C.: Handsome proof-nets: perfect matchings and cographs. Theoretical Computer Science **294**(3), 473–488 (2003). See also the more thorough report [39]

[44] Rival, I. (ed.): Graphs and Order, NATO ASI series C, vol. 147. Kluwer (1985)

[45] Roorda, D.: Resource logic: proof theoretical investigations. Ph.D. thesis, FWI, Universiteit van Amsterdam (1991)

[46] Roorda, D.: Proof nets for Lambek calculus. Logic and Computation **2**(2), 211–233 (1992)

[47] Selinger, P.: Control categories and duality: on the categorical semantics of the lambda-mu calculus. Mathematical Structures in Computer Science **11**(2), 207–260 (2001). URL http://journals.cambridge.org/action/displayAbstract?aid=68983

[48] Slavnov, S.: On noncommutative extensions of linear logic. Logical Methods in Computer Science **Volume 15, Issue 3** (2019). DOI 10.23638/LMCS-15(3:30)2019. URL https://lmcs.episciences.org/5774

[49] Straßburger, L.: Linear logic and noncommutativity in the calculus of structures. Ph.D. thesis, Technische Universität Dresden (2003)

[50] Yetter, D.N.: Quantales and (non-commutative) linear logic. Journal of Symbolic Logic **55**, 41–64 (1990)