Connection-based proof construction in Non-Commutative Logic

D. Galmiche and J.-M. Notin

LORIA - Université Henri Poincaré
54506 Vandœuvre-lès-Nancy Cedex, France
galmiche{notin}@loria.fr

Abstract. We propose a connection-based characterization of the multiplicative fragment of non-commutative logic (MNL), that is a conservative extension of both commutative (MLL) and non-commutative or cyclic (MCyLL) linear logic. It is based on a characterization for MLL together with the introduction of labels and constraints from a formula polarization process. We also study a similar characterization for the intuitionistic fragment of MNL. Finally, we consider the relationships between these results and proof nets construction in MNL based on labels and constraints.

1 Introduction

Since last years there is an increasing amount of interest for logical systems that are resource sensitive (linear logic [10] with its resource consumption interpretation and bunched implications logic [16] with its resource sharing interpretation) but also order-aware (non-commutative logic [1]). As specification logics, they can represent features as interaction, resource distribution and mobility, non-determinism, sequentiality or coordination of entities. Here, we focus on this order-awareness in the Non-Commutative Logic (NL) [1, 17] that incorporates, in a conservative way, the notion of order into Linear Logic. This logic is motivated by a fine logical characterization of operational aspects of concurrent constraint programming (CC) [18]. It allows to analyze how to manage coordination in the access to resources and information [3]. Its intuitionistic fragment also allows to specify and verify the TCP/IP protocol [12] for which one needs a linear implication that consumes its premise in a particular order, together with a commutative conjunction.
In this paper we focus on proof construction in the multiplicative fragment of Non-Commutative Logic (MNL), that is a conservative extension of both commutative (MLL) and non-commutative or cyclic (MCyLL) linear logic. Non-Commutative Logic is mainly characterized by two equivalent sequent calculi that are enriched with a structure of order, the first with an order variety [1] and the second with bunches of formulae including a commutative and a non-commutative separators [17]. Proof search is difficult in such settings in particular because of the management of context splitting. Recent works on MNL proof

search are based on particular treatments of order varieties and the introduction of the focusing principle. We can mention a cluster calculus with a simple way to propagate the order structure [4] and a bipolar sequent calculus that allows to deal with partial information, through specific constraints, during proof construction [2]. The later approach has been also adapted to the proof net construction [3]. Independently, we have studied proof construction in MNL through the construction of labelled proof nets [7], the labels allowing to fix constraints and/or to propagate information necessary to guide and control the construction process. But the labels were not clearly related to the MNL semantics and cannot help for instance for countermodel generation. We recently proposed new sequent and proof net calculi with dependency relations (on formulae) and in this setting the provability of a MNL formula A is studied through the provability of the corresponding MLL formula (A without distinction between commutative and non-commutative connectives) and the verification of some conditions on a set of dependency relations generated during the proof search process [8]. All these works come from, or are influenced by, the definition of NL with order varieties. An alternative way consists in studying proof search from the MNL sequent calculus with bunches [17] and its phase semantics. Because of the presence of two context separators and two structural rules, this calculus is not well adapted to the design of an automatic theorem prover of MNL.
Therefore, we propose a connection-based characterization of provability in MNL, starting from our previous works on connection methods for MLL and proof nets construction [5]. It is based on the introduction of polarities on formulae, of labels and label constraints in order to deal with bunched contexts and thus to capture the specific interactions between commutative and non-commutative connectives. The use of labels and constraints for designing a connection method has been already studied in the setting of a bunched logic, namely BI logic [16], in which labels and constraints are counterparts of the resource semantics [6]. Here an essential preliminary step is the computation of a polarization, which implicitly allows us to consider an intuitionistic encoding of the formula to prove. Moreover, we can naturally define a similar connection-based characterization of provability for the intuitionistic fragment of MNL (IMNL). A consequence of both characterizations is a new characterization of proof nets for both MNL logics and a new procedure for MNL proof nets construction that can be considered as an alternative connection method [5] and a verification method for MNL proof nets [14].

2 The Non-Commutative Logic

The MNL formulae are built from literals $p, q, \ldots, p^\perp, q^\perp, \ldots$ with commutative connectives (conjunction $\otimes$ and disjunction $⅋$) and non-commutative connectives (conjunction $\odot$ and disjunction $\lhd$). The negation is defined as follows with the De Morgan laws: $(p)^\perp = p^\perp$, $(p^\perp)^\perp = p$, $(A \odot B)^\perp = B^\perp \lhd A^\perp$, $(A \lhd B)^\perp = B^\perp \odot A^\perp$, $(A \otimes B)^\perp = B^\perp ⅋ A^\perp$, $(A ⅋ B)^\perp = B^\perp \otimes A^\perp$. The MLL (resp. MCyLL) formulae are built from literals and the connectives $\otimes$ and $⅋$ (resp. $\odot$ and $\lhd$). The

$$\frac{}{\vdash A, A^{\perp}}\ (Ax.) \qquad \frac{\vdash \Gamma, A \quad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}\ (Cut) \qquad \frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Delta, \Gamma, A \otimes B}\ (\otimes)$$

$$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash (\Delta; \Gamma), A \odot B}\ (\odot) \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\ (\parr) \qquad \frac{\vdash \Gamma, (A; B)}{\vdash \Gamma, A \lhd B}\ (\lhd)$$

$$\frac{\vdash \Gamma[\Delta; \Sigma]}{\vdash \Gamma[\Delta, \Sigma]}\ (Entropy) \qquad \frac{\vdash \Gamma, \Delta}{\vdash \Gamma; \Delta}\ (Seesaw) \qquad \frac{\vdash \Gamma; \Delta}{\vdash \Gamma, \Delta}\ (CoSeesaw)$$

Fig. 1. A sequent calculus for MNL

MNL sequent calculus, given in Figure 1, is based on bunches with two formula separators (a commutative ‘,’ and a non-commutative ‘;’). Moreover, it includes some explicit structural rules (Entropy and Seesaw) that establish relationships between both separators. We have added a restricted case of entropy, namely the rule *CoSeesaw*, that is the opposite of the *Seesaw* rule. More details are given in [17]. The phase semantics of MNL is defined as follows:

Definition 1 (Non-commutative phase space). *A* non-commutative phase space *is a sextuplet* $\mathcal{P} = (P, \cdot, \star, 1, \leq, \perp)$ *such that:*
- $(P, \cdot, 1)$ *is a monoid,*
- $(P, \star, 1)$ *is a commutative monoid,*
- $\leq$ *is a partial order on* P*, compatible with both monoidal structures and such that:* $\forall x, y \in P\ x \star y \leq x \cdot y$*,*
- $\perp \subseteq P$ *is an order ideal such that:* $\forall x, y \in P\ x \cdot y \in \perp \Leftrightarrow x \star y \in \perp$*.*

Given a phase space $\mathcal{P}$, $F \subseteq P$ and $G \subseteq P$, we define the following operations: $F \otimes G = (F \star G)^{\perp\perp}$; $F \odot G = (F \cdot G)^{\perp\perp}$; $F \parr G = (G^{\perp} \star F^{\perp})^{\perp}$ and $F \lhd G = (G^{\perp} \cdot F^{\perp})^{\perp}$

Definition 2 (Validity). *Given a phase space* $\mathcal{P}$ *and a valuation* S *that assigns a fact of* P *to any propositional symbol, we define the* interpretation $[\![\cdot]\!]_S^P$ *as usual. A formula* F *is* valid *if and only if* $1 \in [\![F]\!]_S^P$ *for every phase space* $\mathcal{P}$ *and every interpretation* S*.*

Theorem 1. *The sequent calculus with explicit structural rules is sound and complete w.r.t. the phase semantics.*

Proof. The complete proof is given in [17].

We aim to start from the sequent calculus with bunches and its proof-theoretical results in order to study proof search in MNL. Because of the presence of two context separators and two structural rules, this calculus is not well adapted for the design of proof search methods. For instance, the context splitting problem is hardened by the structure of sequents (see $\otimes$ and $\odot$ rules).
The structural rules impose an order between the sequent formulae modulo cyclicity. There is no explicit rule for cyclicity, but it can be recovered through

applications of *Seesaw* and *Entropy* rules. In fact, in that case, we consider a restricted case of entropy through the rule *CoSeesaw* (see Figure 1).
As an example, we consider the sequent $\vdash A, (B; (C, D))$. It can be derived from $\vdash (A; B), C, D$ by a reversible sequence of *Seesaw* and *Entropy* rules. Both sequents can be considered to be equivalent modulo cyclicity.
In order to represent sequents that are equivalent modulo reversible applications of *Seesaw* and *CoSeesaw* rules (ie modulo cyclicity), we consider sequents through a "viewpoint". Considering a family $\mathcal{F}$ of sequents that are equivalent modulo cyclicity, we represent $\mathcal{F}$ by $\vdash \Gamma, A$ that is an element of $\mathcal{F}$ with A an arbitrary formula. Thanks to the following theorem any formula of any sequent of $\mathcal{F}$ can be considered as a "viewpoint".

Theorem 2. *Let $\vdash \Gamma$ be a MNL sequent and A be a formula in Γ, we can built from Γ a reversible derivation of $\vdash \Delta, A$ by using only Entropy and Seesaw rules.*

Proof. Given in detail in [17].

Sequent calculi without structural rules have been proposed for MNL [17]. The structure of formulae in such sequents is expressed through *order varieties*, that are a refinement of cyclic orders. Proof search in these calculi has been studied through the focalization and bipolar calculus [2, 4].
Our aim is to propose proof search methods based on simple (algorithmic) structures, that can be easily handled in an automated proof search procedure. In this perspective, order varieties (and their operations) as mathematical structures seem not to be appropriate for the design of useful proof search methods for MNL. Our previous works based on sequents and proof nets with formulae dependencies [7, 8] were a first step to find alternative and useful semantical structures. A key point of our approach is to separate in a simple way the linearity and non-commutativity aspects.

3 A matrix characterization for MNL

We define a matrix characterization for MNL by extending one characterization for MLL [5, 13] with specific labels and constraints that capture the interactions between commutative and non-commutative connectives. In fact, the label calculus allow to express in a simple way constraints on formulae ordering inside a context.

3.1 Dealing with linearity

First, we consider the matrix characterization for MLL of [5], by ignoring the structure of the context, and by seeing $\odot$ as $\otimes$, and $\lhd$ as $⅋$. As usual in matrix characterizations, we first define the formula tree and thus consider the decomposition tree of the formula F to prove, where nodes are called positions (denoted $a_0, a_1, \ldots$). A position u exactly identifies a subformula of F. We identify two

a_0	F_0
a_1	$(B^\perp \parr C^\perp) \odot D^\perp$
a_2	$B^\perp \parr C^\perp$
a_3	$B^\perp$
a_4	$C^\perp$
a_5	$D^\perp$
a_6	$(D \lhd [(C \otimes B) \odot A]) \parr A^\perp$
a_7	$D \lhd [(C \otimes B) \odot A]$
a_8	D
a_9	$(C \otimes B) \odot A$
a_{10}	$C \otimes B$
a_{11}	C
a_{12}	B
a_{13}	A
a_{14}	$A^\perp$

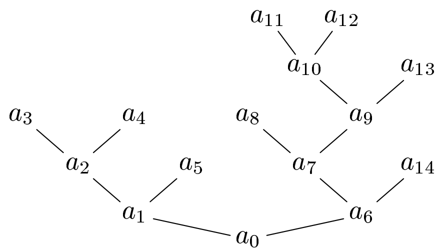

Fig. 2. An example of formula tree

types of nodes, α and β: a node is of type α (resp. β) if the principal connective is $\parr$ or $\lhd$ (resp. $\otimes$ or $\odot$). Now, we define the notion of a path through a formula F, knowing that u denotes a position and $l(u)$ (resp. $r(u)$) denotes the left (resp. right) premise of u.

Definition 3. *Let F be a MNL formula, the set of paths through F is the smallest set of positions such that:*
- $\{a_0\}$ is a path;
- if $\{s, u\}$ is a path then:
if type$(u) = \alpha$ then $\{s, l(u), r(u)\}$ is a path;
if type$(u) = \beta$ then $\{s, l(u)\}$ and $\{s, r(u)\}$ are paths.
An atomic path is a path that contains only atomic positions.

Definition 4. *Given a MNL formula F, a set of connections $\mathcal{SC}$ linearly spans F if and only if*
- all literals in the formula tree appear exactly once in $\mathcal{SC}$;
- if (a, b) and (c, d) are distinct elements of $\mathcal{SC}$ then $a \neq c$ and $b \neq d$;
- every atomic path contains a connection from $\mathcal{SC}$;
- no proper subset of $\mathcal{SC}$ spans F.

This definition is the one for MLL [5] without distinction between commutative and non-commutative connectives in the construction process of atomic paths. Thus, if a MNL formula F is such that there exists a set of connections $\mathcal{SC}$ that linearly spans F then the commutative version of F is provable in MLL.
In order to illustrate the different steps of the characterization, we consider the following MNL formula: $F_0 = (B^\perp \parr C^\perp) \odot D^\perp \parr (D \lhd [(C \otimes B) \odot A]) \parr A^\perp$.
From this first step we obtain the indexed formula tree of F_0 (see Figure 2) and build the set of paths following Definition 3. The set of atomic paths of F_0 is:

$$\left\{ \begin{array}{l} \{a_3, a_4, a_8, a_{11}, a_{14}\}, \{a_3, a_4, a_8, a_{12}, a_{14}\}, \{a_3, a_4, a_8, a_{13}, a_{14}\}, \\ \{a_5, a_8, a_{11}, a_{14}\}, \{a_5, a_8, a_{12}, a_{14}\}, \{a_5, a_8, a_{13}, a_{14}\} \end{array} \right\}$$

The set of connections $\{(a_{14}, a_{13}), (a_3, a_{12}), (a_4, a_{11}), (a_8, a_5)\}$ linearly spans F_0.

$$\frac{x \leq y \quad x' \leq y'}{x \star x' \leq y \star y'}\,(Comp_\star) \qquad \frac{x \leq y \quad x' \leq y'}{x \cdot x' \leq y \cdot y'}\,(Comp.)$$

$$\frac{}{x \star y \leq x \cdot y}\,(Ent) \qquad \frac{}{x \leq x}\,(Ax) \qquad \frac{x \leq y \quad y \leq z}{x \leq z}\,(Tr)$$

Fig. 3. Inference rules for constraint resolution

3.2 Dealing with the non-commutativity

In order to deal with ordering constraints within subformulae, we define label and constraints for MNL that in fact capture the semantics and allow us to encode the structure of MNL sequents, and to express constraints on it. Labels are computed w.r.t. a polarization of the formula that informally defines a "viewpoint" of a sequent.

Definition 5. *A* polarization *pol for a MNL formula F is a function that associates a polarity (+ or −) to F and all its subformulae.*

Definition 6. *Given a formula F and a polarization pol for F, pol is* node-correct *if and only if, for every subformula of F it respects the following inductive rules:*

$$pol(A \otimes B) = + \Rightarrow pol(A) = pol(B) = +;\ pol(A \otimes B) = - \Rightarrow pol(A) \neq pol(B);$$
$$pol(A \odot B) = + \Rightarrow pol(A) = pol(B) = +;\ pol(A \odot B) = - \Rightarrow pol(A) \neq pol(B);$$
$$pol(A \parr B) = - \Rightarrow pol(A) = pol(B) = -;\ \ pol(A \parr B) = + \Rightarrow pol(A) \neq pol(B);$$
$$pol(A \lhd B) = - \Rightarrow pol(A) = pol(B) = -;\ pol(A \lhd B) = + \Rightarrow pol(A) \neq pol(B);$$

Definition 7. *Given a formula F, a set of connections $\mathcal{SC}$ and a polarization pol, pol is* admissible *if and only if it is node-correct and for every connection $(a, b) \in \mathcal{SC}$, $pol(a) \neq pol(b)$.*

Theorem 3. *Given a formula F, if there exists a set of connections $\mathcal{SC}$ that linearly spans F then there exists an admissible polarization for F.*

Proof. Following [15], we can give a linear algorithm that, given a MNL formula, a set of connections and a DR-switching, builds a polarization for F.

Given a polarization of a formula to prove F, we define the notion of label. A label structure is a tuple $\mathcal{L} = (\Sigma, \star, \cdot, 1)$ such that: Σ is a set of so called simple labels, denoted $a, b, c, \ldots$; $\star$ is a commutative monoidal operator; $\cdot$ is a monoidal operator; 1 is a unit for both $\star$ and $\cdot$.
Labels are considered modulo associativity of $\star$ and $\cdot$, and commutativity of $\star$.
A label is said *linear* when it contains no repetition of the same simple label.
We express constraints on the structure of contexts through label constraints of the form $x \leq y$ and the resolution of such constraints is performed using the inference rules given in Figure 3. Given a set of constraints $\mathcal{H}$ and a constraint E, we write $\mathcal{H} \approx\!\!\!| E$ to express that there exists a derivation tree starting from

u	$type(u)$	$lbl(u)$	$lbl(l(u))$	$lbl(r(u))$	$Cont(u)$
$A^+ \otimes^+ B^+$	β	x	y	z	$x \leq y \star z$
$A^+ \otimes^- B^-$	β	l	x	$x \star l$	
$A^- \otimes^- B^+$	β	l	$x \star l$	x	
$A^+ \odot^+ B^+$	β	x	y	z	$x \leq y \cdot z$
$A^+ \odot^- B^-$	β	l	x	$l \cdot x$	
$A^- \odot^- B^+$	β	l	$x \cdot l$	x	
$A^+ \invamp^+ B^-$	α	x	$b \star x$	b	
$A^- \invamp^+ B^+$	α	x	a	$a \star x$	
$A^- \invamp^- B^-$	α	l	a	b	$l \leq b \star a$
$A^+ \lhd^+ B^-$	α	x	$x \cdot b$	b	
$A^- \lhd^+ B^+$	α	x	a	$a \cdot x$	
$A^- \lhd^- B^-$	α	l	a	b	$l \leq b \cdot a$

Fig. 4. Inductive rules for label computation

E, by using inferences given in Figure 3 and such that each branch ends with a Ax-rule, a Ent-rule or with a constraint of $\mathcal{H}$. If $\mathcal{E}$ is a set of constraints, we write $\mathcal{H} \approx \mathcal{E}$ to express that, for all $E \in \mathcal{E}$ we have $\mathcal{H} \approx E$.
Given a MNL formula F and a node-correct polarization for F, we can compute the polarities of F subformulae by starting from F with the polarity + and the label 1 and by applying the rules given in Figure 4.
The labels associated with premises of $\otimes^+$ or $\odot^+$ nodes and to the positive premise of $\otimes^-$ or $\odot^-$ nodes are considered as *variables*. The labels associated with premises of $\invamp^-$ or $\lhd^-$ nodes, and to the negative premise of $\invamp^+$ or $\lhd^+$ nodes are considered as *constants* (simple labels). A substitution σ is an application that associates a label l to a variable x such that l is a element of the label structure $(\mathcal{C}, \star, \cdot, 1)$, where $\mathcal{C}$ denotes the set of constants.
In some cases, we generate label constraints that are splitted into two sets: the set of *assertions* (Ass) that are introduced by $\invamp^-$ or $\lhd^-$ nodes, and the set of *requirements* (Req) that are introduced by $\otimes^+$ and $\odot^+$ nodes. The assertions are axioms in the calculus on constraints.
We can notice that label constraints express some constraints on the formulae structure inside a sequent. Therefore, one could expect to have also such constraints associated to $\invamp^+$, $\lhd^+$, $\otimes^-$ and $\odot^-$. For these nodes, constraints are expressed through the way the premise labels are computed and the constraints derived from connections.

Definition 8. *A formula F of MNL is connection-provable if and only if*
i) there exists a set of connections $\mathcal{SC}$ such that linearly spans F;
ii) there exists an admissible polarization pol and a substitution σ such that:
- *For every connection (a^+, a^-) of $\mathcal{SC}$, $\sigma(Ass) \approx \sigma(a^+ \leq a^-)$,*
- *$\sigma(Ass) \approx \sigma(Req)$.*

Coming back to our example, we can now enrich its formula tree with labels. The polarity and the label associated to each subformula of F_0 are given in the labelled formula tree of F_0 are given in Figure 5. Moreover we have some

u	Pol(u)	lbl(u)	Cont(u)
a_0	$+$	1	
a_1	$-$	e	
a_2	$-$	$s \cdot e$	$s \cdot e \leq b \star c$
a_3	$-$	b	
a_4	$-$	c	
a_5	$+$	s	
a_6	$+$	e	
a_7	$+$	$a \star e$	
a_8	$-$	d	
a_9	$+$	$d \cdot (a \star e)$	$d \cdot (a \star e) \leq t \cdot x$
a_{10}	$+$	t	$t \leq z \star y$
a_{11}	$+$	z	
a_{12}	$+$	y	
a_{13}	$+$	x	
a_{14}	$-$	a	

Fig. 5. An example of labelled formula tree

label constraints. Among them, we can distinguish an assertion: $s \cdot e \leq b \star c$, two requirements: $d \cdot (a \star e) \leq t \cdot x$ and $t \leq z \star y$ and constraints coming from connections: $x \leq a$, $y \leq b$, $z \leq c$ and $s \leq d$. In these constraints, $\{a, b, c, d, e\}$ are constants (simple labels), and $\{x, y, z, s, t\}$ are variables.

In order to solve these constraints, we can easily deduce, from the constraints obtained from connections, a partial substitution: $\{x \mapsto a, y \mapsto b, z \mapsto c, s \mapsto d\}$. Moreover, the requirement $t \leq z \star y$ becomes $t \leq c \star b$ and thus can be satisfied if we consider $t \mapsto b \star c$. Then, this substitution trivially satisfies constraints from connections and the requirement $t \leq z \star y$. We have now to verify that it also satisfies the last requirement $d \cdot (a \star e) \leq t \cdot x$. Using inference rules of Figure 3 we can obtain the following proof:

$$\dfrac{\dfrac{d \leq d \quad a \star e \leq e \cdot a}{d \cdot (a \star e) \leq d \cdot e \cdot a}\,(Comp) \qquad \dfrac{d \cdot e \leq b \star c \quad a \leq a}{d \cdot e \cdot a \leq (b \star c) \cdot a}\,(Comp)}{d \cdot (a \star e) \leq (b \star c) \cdot a}\,(Tr)$$

Let us remark that the label constraints capture the constraints on the bunched context for rule applications in the MNL sequent calculus. For instance, the constraint $a \star e \leq e \cdot a$ corresponds to the use of the *Entropy* rule in the sequent calculus. Finally, we check that the MNL formula of our example is connection-provable and in fact valid (see next subsection).

Let us consider now the formula $G_0 = (B^\perp \parr C^\perp) \odot D^\perp \parr (D \lhd [(C \otimes B) \otimes A]) \parr A^\perp$ that is obtained from the previous one by changing $(C \otimes B) \odot A$ to $(C \otimes B) \otimes A$. If we take the same polarization as for F_0, we obtain the same computed labels. But the requirement generated by the subformula $(C \otimes B) \otimes A$ is now $d \cdot (a \star e) \leq t \star x$. Using the substitution, we obtain the constraint $d \cdot (a \star e) \leq b \star c \star a$, that cannot be satisfied.

Taking another admissible polarization changes the labels and the constraints associated to subformulae, but the set of constraints does not admit a solution. Then, as we have no other choice for the set of connections, we can state that G_0 is not connection-provable, and consequently is not valid in MNL.

3.3 Properties of the characterization

Let us show that our characterization is correct and complete w.r.t. the MNL sequent calculus with bunches. It is mainly based on: a) a polarization of the sequent in order to deal with irrelevant (reversible) applications of (co-)seesaw rules; b) labels to represent the structure of formulae inside a sequent; c) constraints on labels to deal with applications of the *Entropy* rule in the sequent calculus.

A key point is that provability is independent from the choice of an admissible polarization. If, for such a choice, the set of constraints has no solution then we can claim the non-provability for the given set of connections and thus try another one if possible. An admissible polarization for F is defined in such a way that if $\vdash \Gamma$ is a sequent that can be derived from F, then there exists exactly one positive formula in $\vdash \Gamma$. In some way, our polarization gives a "viewpoint" of a MNL sequent, as defined in the previous section. The structure of formulae within a sequent $\vdash \Delta$ is considered w.r.t. the structure of $\vdash \Gamma^-, A^+$, where $\vdash \Gamma, A$ is equivalent to $\vdash \Delta$ modulo cyclicity. Moreover, labels are computed w.r.t. the polarization in such a way that, given a set of connections $\mathcal{SC}$, the sets of constraints generated from a choice of an admissible polarization are all equivalent. This is related to the fact that, in the MNL sequent calculus, sequents are equivalent modulo cyclicity (as said in previous section). As an example, we consider the part of the proof of F_0 (see Figure 6). Each formula inside a sequent is together with its label. We can easily see that the label $d \cdot (a \star e)$ (associated with $(C \otimes B) \odot A$, that is the positive formula of S_2 and S_2') represents the structure of both S_2 and S_2'.

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\vdash [\mathbf{e} : (B^\perp \parr C^\perp) \odot D^\perp ; \mathbf{d} : D], \mathbf{t} : C \otimes B
\qquad
\dfrac{}{\vdash \mathbf{x} : A, \mathbf{a} : A^\perp}\,(Ax.)
}{
S_1 : \vdash [\mathbf{a} : A^\perp ; \mathbf{e} : (B^\perp \parr C^\perp) \odot D^\perp ; \mathbf{d} : D], \mathbf{d} \cdot (\mathbf{a} \star \mathbf{e}) : (C \otimes B) \odot A
}\,(\odot)
}{
S_2' : \vdash [(\mathbf{a} : A^\perp, \mathbf{e} : (B^\perp \parr C^\perp) \odot D^\perp); \mathbf{d} : D], \mathbf{d} \cdot (\mathbf{a} \star \mathbf{e}) : (C \otimes B) \odot A
}\,(Entropy)
}{
\vdash [\mathbf{a} : A^\perp, \mathbf{e} : (B^\perp \parr C^\perp) \odot D^\perp]; \mathbf{d} : D ; \mathbf{d} \cdot (\mathbf{a} \star \mathbf{e}) : (C \otimes B) \odot A
}\,(Seesaw)
}{
S_2 : \vdash \mathbf{e} : (B^\perp \parr C^\perp) \odot D^\perp, (\mathbf{d} : D ; \mathbf{d} \cdot (\mathbf{a} \star \mathbf{e}) : (C \otimes B) \odot A), \mathbf{a} : A^\perp
}\,(Ent.)
$$

Fig. 6. A part of proof with labels

Now, we illustrate how label constraints are related to ordering constraints within a sequent. Let us consider the sequent $\vdash \Sigma, A \odot B$. In order to apply a $\odot$-rule we need to turn the context Σ into the form $\Delta ; \Gamma$. This transformation is not reversible in general and induces some constraints on (the structure of) Γ and Δ. Let us consider the way labels and constraints are computed for $\odot^+$. The requirement $l \leq x \cdot y$ expresses that Σ has to be turned into the particular form $\Delta ; \Gamma$, and thus some constraints on variables x and y. In the case of $\odot^-$, we have to compute labels in a different way taking into account the polarization. Constraints on contexts are expressed by the way labels are computed and by

equations coming from connections.
In our example, we have to apply an *Entropy* rule to obtain the sequent S_1 from S_2. The structure of S_1 would be represented by the label $d \cdot e \cdot a$. In our characterization, we associate a constraint $d \cdot (a \star e) \leq t \cdot x$ to $(C \otimes B) \odot A$. We have to use an entropy relation namely $a \star e \leq e \cdot a$, to solve this constraint.
Let us remind that a MNL sequent $\vdash \Gamma$ is provable in MNL if and only if $Dis(\Gamma)$ is connection-provable, where $Dis : Context \mapsto Formula$ is defined by: $Dis(A) = A$ if A is a formula; $Dis(\Gamma, \Delta) = Dis(\Gamma) \parr Dis(\Delta)$; $Dis(\Gamma; \Delta) = Dis(\Gamma) \lhd Dis(\Delta)$.

Theorem 4 (Correctness). *If a MNL formula F is connection-provable then it is provable in the MNL sequent calculus.*

Proof. At first, the existence of a set of connections that linearly spans F gives a proof in MLL of the commutative version of F. It provides a skeleton for the proof of F in MNL. Then, we need to attach a structure to each sequent of this skeleton. For that, we use the label associated to each formulae. In fact, the label of the positive formula induces an order between other (negative) formulae. The existence of a solution such that label constraints are satisfied ensures that we can insert structural rules, between each inference of the skeleton, such that the proof construction leads to a MNL proof.

Theorem 5 (Completeness). *If a MNL formula F is provable in the MNL sequent calculus then F is connection-provable.*

Proof. If the formula F is provable, then there exists a proof of F in the MNL sequent calculus. From this proof, we can deduce a set of connections that linearly spans F (see [5] for more details).
We can then compute an admissible polarization for F and by linearity, in any sequent appearing in the proof there is exactly one positive formula (other formulae of the sequent being negative). Thus, the polarization gives a so-called viewpoint in any sequent, and the label associated to the positive formula represents the structure of the negative formulae. Then, given this viewpoint, we can relate the rule of structural entropy with the corresponding *Entropy* rule in the calculus of constraints.

3.4 The related connection method

From this characterization we can define a connection-based proof search method with the following main steps: 1) to compute the atomic paths and choose a spanning set of connections like in MLL [5, 13]; 2) to compute, from the set of connections, an admissible polarity and then labels and constraints; 3) to compute a substitution that is a solution for the set of constraints (connection constraints and requirements satisfied from assertions).
Steps 1) and 2) being standard and without difficulty we aim to analyze the step 3) and its main problem, that is to find a substitution such that requirements can be proved from assertions using some particular inference rules. Given these

rules, constraints of the form $x \leq a$ (where x is a variable and a is a constant) are satisfied if and only if $\sigma(x) = a$. The way in which constraints are generated makes constraint resolution relatively easy. Moreover, we can use the linearity of labels to reduce the search space.

Assertions are of the form $l \leq a \square b$, where a and b are constants (simple labels) and l is a label composed of variables and constants (l does not contain a or b). Requirements are of the form $l \leq x \square y$, where x and y are (fresh) variables. Thus, variables have a very restricted set of possible instantiations. Moreover, labels attached to positive formulae always appear on the left-hand side of a constraint (generated by a connection or by a $\otimes^+$ or $\odot^+$ rule). However, when created, variables are attached to positive premise (for instance, see the construction of labels for $\otimes^+$ in Figure 4). As a result, variables always appear in the right-hand side of a constraint and on the left-hand side of another one. This means that the set of admissible values of each variable is strongly restricted by the constraints. Coming back to our example, the variable t appears on the left-hand side of $t \leq z \star y$ (y and z appear in the constraint $y \leq b$ and $z \leq c$). It also appears on the right-hand side of $d \cdot (a \star e) \leq t \cdot x$. Thus, from this requirement (and taking into account that $\sigma(x) = a$, we can deduce the following constraint $d \cdot e \leq t$, using compatibility and entropy rules. Then, the choices of values for t are strongly reduced because of the constraint $d \cdot e \leq t \leq b \star c$ that admits a solution because we have an assertion $d \cdot e \leq b \star c$. In our example, we have seen that $\sigma(t) = b \star c$ satisfies requirements and that $\sigma(t) = d \cdot e$ is also a solution for t. These two solutions for t can be considered equivalent, as they are deducible from each other using the assertion. The choice between $b \star c$ and $d \cdot e$ depends on strategies applied to compute a substitution. A smart choice of substitution can strongly reduce the complexity of constraints resolution.

4 A Characterization for Intuitionistic MNL

From the previous results on MNL, we aim to analyze what happens in the case of Intuitionistic MNL (IMNL). This work is inspired from the relationships between MLL and MILL described in [15]. IMNL formulae are built from propositional variables $p, q, \ldots$ and commutative connectives $\otimes$ and $\multimap$, and non-commutative ones $\odot$, $-\!\!\bullet$ and $\bullet\!\!-$. The corresponding sequent calculus is given in Figure 7.

There exists a strong relationship between MLL and intuitionistic MLL [15] that can be extended to MNL. In fact, any IMNL formula can be translated to a MNL formula using De Morgan laws and the equalities: $A \multimap B = A^{\perp} \parr B$, $A -\!\!\bullet B = A^{\perp} \lhd B$ and $B \bullet\!\!- A = B \lhd A^{\perp}$. For that, given a IMNL formula A, we define its classical counterpart $\overline{A}$ (and $\underline{A}$).

Let A be an IMNL formula, we define its classical counterpart $\overline{A}$ (and $\underline{A}$) as follows:

$$
\begin{array}{cccc}
\overline{a} = a & \underline{a} = a^{\perp} & \overline{A \otimes B} = \overline{A} \otimes \overline{B} & \underline{A \otimes B} = \underline{B} \parr \underline{A} \\
\overline{A \odot B} = \overline{A} \odot \overline{B} & \underline{A \odot B} = \underline{B} \lhd \underline{A} & \overline{A \multimap B} = \underline{A} \parr \overline{B} & \underline{A \multimap B} = \underline{B} \otimes \overline{A} \\
\overline{A -\!\!\bullet B} = \underline{A} \lhd \overline{B} & \underline{A -\!\!\bullet B} = \underline{B} \odot \overline{A} & \overline{B \bullet\!\!- A} = \overline{B} \lhd \underline{A} & \underline{B \bullet\!\!- A} = \overline{A} \odot \underline{B}
\end{array}
$$

$$
\frac{}{A \vdash A}\,(Ax.) \qquad \frac{\Gamma[\Delta;\Sigma] \vdash C}{\Gamma[\Delta,\Sigma] \vdash C}\,(Ent.) \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\,(\otimes_R)
$$

$$
\frac{\Gamma[A,B] \vdash C}{\Gamma[A \otimes B] \vdash C}\,(\otimes_L) \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma;\Delta \vdash A \odot B}\,(\odot_R) \qquad \frac{\Gamma[A;B] \vdash C}{\Gamma[A \odot B] \vdash C}\,(\odot_L)
$$

$$
\frac{\Gamma \vdash A \quad \Delta[B] \vdash C}{\Delta[\Gamma, A \multimap B] \vdash C}\,(\multimap_L) \qquad \frac{A, \Gamma \vdash B}{\Gamma \vdash A \multimap B}\,(\multimap_R) \qquad \frac{\Gamma \vdash A \quad \Delta[B] \vdash C}{\Delta[\Gamma; A \mathbin{-\!\bullet} B] \vdash C}\,(-\!\bullet_L)
$$

$$
\frac{A;\Gamma \vdash B}{\Gamma \vdash A \mathbin{-\!\bullet} B}\,(-\!\bullet_R) \qquad \frac{\Gamma \vdash A \quad \Delta[B] \vdash C}{\Delta[B \mathbin{\bullet\!-} A; \Gamma] \vdash C}\,(\bullet\!-_L) \qquad \frac{\Gamma; A \vdash B}{\Gamma \vdash B \mathbin{\bullet\!-} A}\,(\bullet\!-_R)
$$

Fig. 7. A sequent calculus for IMNL

From this definition we have the following result

Theorem 6. *A IMNL sequent $H_1, \ldots, H_n \vdash C$ is provable if and only if the sequent MNL $\vdash \underline{H_1}, \ldots, \underline{H_n}, \overline{C}$ is provable.*

Proof. We build a proof of $H_1, \ldots, H_n \vdash C$ from a proof of $\vdash \underline{H_1}, \ldots, \underline{H_n}, \overline{C}$ and conversely. An intuitionistic sequent $\Gamma \vdash A$, where Γ is a context with bunches, can be easily turned into a one-sided sequent, using negation and De Morgan laws. Moreover, as stated in theorem 2, any sequent $\vdash \Delta[A]$ can be reversibly turned into the form $\vdash \Gamma, A$.

Consequently, any linearly balanced IMNL formula A can be transformed to a variant MNL formula which is built from polarized atoms and polarized connectives in the following way

$$
\begin{array}{ccc}
\ulcorner a \urcorner = a^+ & \llcorner a \lrcorner = a^- & \ulcorner A \otimes B \urcorner = \ulcorner A \urcorner \otimes^+ \ulcorner B \urcorner \\
\llcorner A \otimes B \lrcorner = \llcorner B \lrcorner \parr^- \llcorner A \lrcorner & \ulcorner A \odot B \urcorner = \ulcorner A \urcorner \odot^+ \ulcorner B \urcorner & \llcorner A \odot B \lrcorner = \llcorner B \lrcorner \lhd^- \llcorner A \lrcorner \\
\ulcorner A \multimap B \urcorner = \llcorner A \lrcorner \parr^+ \ulcorner B \urcorner & \llcorner A \multimap B \lrcorner = \llcorner B \lrcorner \otimes^- \ulcorner A \urcorner & \ulcorner A \mathbin{-\!\bullet} B \urcorner = \llcorner A \lrcorner \lhd^+ \ulcorner B \urcorner \\
\llcorner A \mathbin{-\!\bullet} B \lrcorner = \llcorner B \lrcorner \odot^- \ulcorner A \urcorner & \ulcorner B \mathbin{\bullet\!-} A \urcorner = \ulcorner B \urcorner \lhd^+ \llcorner A \lrcorner & \llcorner B \mathbin{\bullet\!-} A \lrcorner = \ulcorner A \urcorner \odot^- \llcorner B \lrcorner
\end{array}
$$

The transformation of a one-sided sequent into a two-sided (intuitionistic) sequent corresponds to a polarization of the sequent, as previously defined. In fact, a node-correct polarization follows De Morgan laws, and the definition of linear implication(s). Given a IMNL formula A we built a MNL formula $\ulcorner A \urcorner$ formula with a unique polarization, denoted $ipol(A)$, derived from the above construction. It is easy to verify that it is node-correct. From now, we can apply the connection-based method of MNL and derive the following definition:

Definition 9. *A formula F of IMNL is connection-provable if and only if*
i) there exists a set of connections $\mathcal{SC}$ such that linearly spans $\ulcorner A \urcorner$.
ii) given the polarization $ipol(A)$, there exists a substitution σ such that:
- for every connection (a^+, a^-) of $\mathcal{SC}$, $\sigma(Ass) \models\!\approx \sigma(a^+ \leq a^-)$,
- $\sigma(Ass) \models\!\approx \sigma(Req)$.

It means that a formula A of IMNL is connection-provable if and only if $\ulcorner A \urcorner$ is connection-provable in MNL but with a fixed polarization $ipol(A)$.

Theorem 7. *A IMNL formula A is valid if and only if A is connection-provable.*

Proof. Similar to the proof for MNL with $ipol(A)$ as an admissible polarization.

As an example, we can consider $A, D \multimap\!\!\bullet (C \otimes B) \vdash D \multimap\!\!\bullet [(C \otimes B) \odot A]$. Its translation into an equivalent polarized MNL formula leads to the formula F_0 with the polarization given in Figure 5.

5 MNL Proof nets

The notion of *proof net* has been introduced by Girard [11] in order to cope with the problems due to the intrinsic concurrence of the linear sequent calculus. It corresponds to a concise graph-theoretic representation of linear logic deductions. This concept, initially defined for MLL, has been defined for various LL fragments and then for MNL [1]. We can define a proof structure from a pair $(Dectree, Axlinks)$ where, for a given sequent, $Dectree$ is the set of its subformulae arranged as a decomposition tree and $Axlinks$ is a set of axiom-links (or connections) between positive and negative occurrences of the same atomic formula. Such a pair is a proof structure if each atom of $Dectree$ is the conclusion of exactly one axiom-link.
There exist various correctness criteria that decides if a proof structure of a LL fragment corresponds to a (graphical) proof of a given sequent or not, i.e., if a proof structure is a proof net. It is known that a based-on connection characterization of MLL directly provides a new correctness criterion for MLL proof nets [5]. Moreover, an algorithm that builds a MLL proof net provides a connection method that builds in parallel a sequent proof and a proof net, in case of provability [5, 9].

Correctness and verification of proof nets.
As a first consequence of our based-on connection characterization of MNL we can straightforwardly deduce a new correctness criterion for MNL proof nets based on a criterion for MLL with the computation of a polarization, the generation of labels and constraints and the satisfaction of requirements from assertions. Thus, we can design an algorithm for the verification of a MNL proof structure, starting by the verification of the MLL proof structure associated to the MNL one (without difference between commutative and non-commutative connectives), for instance with a linear algorithm [15]. The proof structure including a given set of connections, we can compute in parallel an admissible

polarization for it (following [15]), and then compute labels attached to nodes (subformulae) of the proof structure and the set of constraints to satisfy.

Proof nets construction.
Given a MLL formula F, there exists an algorithm [5] that automatically builds a set of axiom-links (or connections) such that the decomposition tree of F, together with this set of axiom-links is a proof net. It directly provides a particular connection method for MLL. We can easily adapt this algorithm to compute a set of connections that linearly spans a given MNL formula F. Then, as for the verification, we compute, in parallel with the construction of axiom-links, an admissible polarization for F and then generate the labels attached to subformulae (or nodes) in the proof structure. Finally, we solve the requirements from the assertions and check if there is a solution for the set of constraints. This algorithm can be adapted to intuitionistic MNL proof nets by only checking polarities during the axiom-links construction.

6 Conclusion and Perspectives

We present a connection-based characterization of the multiplicative fragment of Non-Commutative logic (MNL). It is related to a MNL sequent calculus with bunches and its phase semantics [17]. Its main steps are: to consider a MNL formula, first without distinction between commutative and non-commutative connectives; to introduce a polarization, that means to consider MNL through its intuitionistic fragment; to build labels and constraints that are syntactic counterparts of the semantics; to characterize provability through the resolution of constraint. Because of the central role of polarization, we can naturally derive a similar connection-based characterization for the intuitionistic fragment of MNL. Compared to other works on MNL proof construction, that are based on order varieties and the focusing principle [2, 4] and that deal with partial information in a specific way, our approach tends to separate the linearity from the non-commutativity during the proof construction process with the help of labels, constraints and constraint resolution. A similar approach has been already studied for the BI logic [16], in which additive and multiplicative connectives cohabit, but with different labels and constraints that capture a different semantics [6]. Moreover, we can derive a new correctness criterion for MNL proof nets and propose algorithms for the verification and the construction of MNL proof nets from our previous works on MLL [5].
In further work, we will focus on the constraint resolution process that deals with simple labels and particular constraints. Moreover, we will study the complexity of MNL proof net verification from our new criterion, in comparison with the quadratic one of [14]. The labels and constraints being syntactic counterparts of the semantics, we will also consider the problem of countermodel generation in MNL in case of non-provability.

References

1. M. Abrusci and P. Ruet. Non-commutative logic I : the multiplicative fragment. *Annals of Pure and Applied Logic*, 101:29–64, 2000.
2. J.M. Andreoli. Focussing and Proof construction. *Annals of Pure and Applied Logic*, 107:131–163, 2001.
3. J.M. Andreoli. Focusing proof-net construction as a middleware paradigm. In *18th Int. Conference on Automated Deduction, CADE-18, LNAI 2392*, pages 501–516, 2002. Copenhagen, Danemark.
4. C. Faggian. Proof construction and non-commutativity: a cluster calculus. In *International Conference on Principles and Practice of Declarative Programming, PPDP 2000*, 2000. ACM Press.
5. D. Galmiche. Connection Methods in Linear Logic and Proof nets Construction. *Theoretical Computer Science*, 232(1-2):231–272, 2000.
6. D. Galmiche and D. Méry. Connection-based proof search in propositional BI logic. In *18th Int. Conference on Automated Deduction, CADE-18, LNAI 2392*, pages 111–128, 2002. Copenhagen, Danemark.
7. D. Galmiche and J.M. Notin. Proof-search and Proof nets in Mixed Linear Logic. *Electronic Notes in Theoretical Computer Science*, 37, 2000.
8. D. Galmiche and J.M. Notin. Calculi with Dependency Relations for Mixed Linear Logic. In *International Workshop on Logic and Complexity in Computer Science, LCCS'2001*, pages 81–102, Créteil, France, 2001.
9. D. Galmiche and G. Perrier. A procedure for automatic proof nets construction. In *LPAR'92, International Conference on Logic Programming and Automated Reasoning, LNAI 624*, pages 42–53, St. Petersburg, Russia, July 1992.
10. J.Y. Girard. Linear Logic: its Syntax and Semantics. In J.Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, pages 1–42. Cambridge University Press, 1995.
11. J.Y. Girard. Proof nets: the parallel syntax for proof theory. In A. Ursini and P. Agliano, editors, *Logic and Algebra*, New York, 1995. M. Dekker.
12. D. Gray, G. Hamilton, J. Power, and D. Sinclair. Specifying and verifying TCP/IP using Mixed Intuitionistic Linear Logic. Technical report, Dublin City Univ., 2001.
13. C. Kreitz, H. Mantel, J. Otten, and S. Schmitt. Connection-based proof construction in Linear Logic. In *14th Int. Conference on Automated Deduction, CADE-14, LNCS 1249*, pages 207–221, Townsville, North Queensland, Australia, 1997.
14. V. Mogbil. Quadratic correctness criterion for non commutative logic. In *15th Int. Workshop on Computer Science Logic, CSL 2001, LNCS 2142*, pages 69–83, Paris, France, 2001.
15. A. S. Murawski and C.-H. L. Ong. Dominator trees and fast verification of proof net. In *15th IEEE Symposium on Logic in Computer Science*, Santa Barbara, California, June 2000.
16. P.W. O'Hearn and D. Pym. The Logic of Bunched Implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.
17. P. Ruet. Non-commutative logic II : sequent calculus and phase semantics. *Math. Struct. in Comp. Science*, 10:277–312, 2000.
18. P. Ruet and F. Fages. Concurrent constraint programming and non-commutative logic. In *11th Int. Workshop on Computer Science Logic, CSL'97, LNCS 1414*, pages 406–423, Aarhus, Denmark, August 1997.