Structures for Multiplicative Cyclic Linear Logic: Deepness vs Cyclicity *

Pietro Di Gianantonio

dipartimento di Matematica e Informatica, Università di Udine
via delle Scienze 206 I-33100, Udine – Italy
e-mail: digianantonio@dimi.uniud.it

Abstract. The aim of this work is to give an alternative presentation for the multiplicative fragment of Yetter's cyclic linear logic. The new presentation is inspired by the calculus of structures, and has the interesting feature of avoiding the cyclic rule. The main point in this work is to show how cyclicity can be substituted by deepness, i.e. the possibility of applying an inference rule at any point of a formula. We finally derive, through a new proof technique, the cut elimination property of the calculus.

keywords: proof theory, linear logic, cyclic linear logic, calculus of structures, Lambek calculus.

1 Introduction

A non-commutative version of linear logic appeared as soon as linear logic was published [1]; in 1987 Jean Yves Girard, in a series of lectures, suggested a version of linear logic containing non-commutative connectives. This logic was later fully developed by Yetter [2] and named Cyclic Linear Logic (CyLL). This immediate interest for a non-commutative logic can be explained by the fact that linear logic puts great emphasis on structural rules, and so it was natural to consider the commutativity rule and check whether it is possible to define a proof system without it. Looking at the subject from a semantic point of view, non-commutative connectives are present in the "logic of quantum mechanics" [3] a logic aiming to model empirical verification and containing a non commutative connective "and then" (&). In this logic, the formula $A\&B$ is interpreted as "we have verified A and then we have verified B". Non-commutative connectives are present also in Lambek's syntactic calculus [4], a calculus modeling linguistic constructors. Both these calculi are strictly related with cyclic linear logic, in particular, Lambek's calculus can be seen as a fragment of the multiplicative cyclic linear logic. Later on the cyclic linear logic has been extended by the introduction of a commutative version of the multiplicative connectives [5, 6], leading to the definition of non-commutative logic (NL), a logic that encompasses both cyclic linear logic and standard linear logic.

* Supported by Italian MIUR Cofin "Protocollo", and EEC Working Group "Types"

The multiplicative fragment of cyclic linear logic can be simply obtained by taking the multiplicative LL, and substituting the structural rule of Exchange:

$$\frac{\Delta}{\Gamma} \quad \text{the list } \Gamma \text{ is a permutation of the list } \Delta$$

with the Cycling rule:

$$\frac{\Delta, \Gamma}{\Gamma, \Delta} \quad \text{Cycling}$$

Cycling rule is considered so crucial that the whole logic is named after it. However this rule still misses a natural explanation, and it is often explained in terms of necessity:

> The reader should note that in terms of the semantics we will develop, the seemingly unnatural Cycling rule is forced by having a system with a single negation ... [2]

More recently Guglielmi proposed the calculus of structures (CoS) [7] as a calculus for defining logics, alternative to sequent calculus and whose main feature is deep inference, that is the possibility of applying inference rules arbitrarily deep inside formulae. This greater liberty in applying inference rules can be used to treat logics whose formalization in the sequent calculus is not completely satisfactory (as modal logic [8]), or to ensure structural properties for the derivation, properties that are not present in sequent calculus derivations (as locality [9, 10]). Moreover there are examples of logics that cannot be treated at all in sequent calculus [7].

In this work, we present cyclic linear logic using the CoS. We show that the cycling rule can be avoided in the CoS formulations, namely, we show that if one takes the formulation of multiplicative linear logic in the CoS and then simply drops the commutative rules for par and tensor, one immediately obtains a formulation of cyclic linear logic (with no cycling rule present). This fact gives an explanation for the cycling rule, i.e. the cycling rule is a rule that recover the lack of deep inference in the sequent calculus. More in detail, deep inference can be, informally, described as follows:

> for any formulae A, B and positive context S, from $A \Rightarrow B$ and $S[A]$ derive $S[B]$.

Deep inference as a rule is normally not present in proof systems but one can argue, after having defined positive contexts, that deep inference has to be an admissible rule. In our work, we show that, with respect to admissibility, the cycling rule and deep inference are equivalent, i.e., for any proof system containing all the remaining rules of cyclic linear logic, the cycling rule is admissible if and only if deep inference is admissible. Therefore, the CoS gives a method for substituting, in Yetter's words, a *seemingly unnatural rule*: cycling, with a natural one: deep inference. The proof transformation between the two systems

does not add complexity cost: given a CyLLproof in sequents calculus, it is possible to define a corresponding proof in the CoS containing the same order of applied rules.

As a further result, in this work we present a new technique for proving cut elimination that can be usefully employed in the treatment of other logics inside the CoS. The CoS is a very recent formalism and so there is the need of developing a bunch of techniques for obtaining meta-theoretical results. Our original proof of cut elimination is a step in this direction.

The article is organized as follows: Section 2 gives a short explanation of the calculus of structures, Section 3 formalizes the multiplicative linear logic in the CoS, Section 4 presents cyclic linear logic together with a proof of cut elimination, Section 5 gives a short account for possible future works.

2 Calculus of Structures

The CoS is characterized by two main features, the possibility of applying inference rules at any point in a formula (deep inference) and the idea to consider formulae up to an equivalence relation equating formulae provable equivalent by some elementary arguments. The equivalence classes of this relation are called *structures*. In this article we retain the first feature of the CoS but drop the second one, i.e. we do not use structures and work directly on formulae. The main reason for this choice is the fact that the cut elimination proof, given in following, needs to consider a proof system where also formulae belonging to the same equivalence class are kept distinct. As a consequence, we do not present here the true CoS but a slightly different formalism using a different syntax and having a different treatment of the equality. The reader not familiar with the CoS will gain a more direct presentation: we use only formulae and we avoid the syntactic overhead caused by structures. The reader familiar with the CoS should have no problem in relating the two presentations.

Before introducing our formalism, we want to present an alternative view of the sequent calculus. In constructing a derivation for a formula A, in sequent calculus, one reduces the derivability of a formula A to the derivability of a set of sequents. That is, in the intermediate steps of the construction of a bottom up derivation of A, one reduces the problem of deriving A to the problem of deriving a set of sequents. The intuitive meaning of this set of sequents (let it be $\vdash B_{1,1}, \ldots, B_{1,n_1} \quad \vdash B_{2,1}, \ldots, B_{2,n_2} \quad \ldots \quad \vdash B_{m,1}, \ldots, B_{m,n_m}$) is the formula $(B_{1,1} \parr \ldots \parr B_{1,n_1}) \otimes (B_{2,1} \parr \ldots \parr B_{2,n_2}) \otimes \ldots \otimes (B_{m,1} \parr \ldots \parr B_{m,n_m})$. In fact one can easily map a set of derivations for the sequents $\vdash B_{1,1}, \ldots, B_{1,n_1}$ $\vdash B_{2,1}, \ldots, B_{2,n_2} \quad \ldots \quad \vdash B_{m,1}, \ldots, B_{m,n_m}$, into a derivation for the formula and vice-versa. One can see the above set of sequents as a different writing of the previous formula. This alternative writing of a formula is a way of marking the main connectives to which inference rules can be applied. We argue that sequent calculus is a formalism for writing derivations where the main connectives have, at most, syntactic deepness two. In this respect, the most remarkable difference

between sequent calculus and the CoS lies on the fact that in the CoS rules can be applied at an arbitrary deepness inside a formula.

In the CoS there is more freedom in applying rules, and as a consequence, derivations loose some of their internal structures but on the other hand there are many examples of logics where the use of deep rules gives some advantages from the point of view of the proof theory [9, 7, 8, 10].

3 Multiplicative Linear Logic

As a first step in presenting cyclic linear logic, we present multiplicative linear logic (MLL) [1] in the CoS. We use the standard syntax of multiplicative linear logic, i.e. our formulae are given by the syntax:

$$A := \bot \mid \mathbf{1} \mid a \mid \overline{a} \mid (A \parr A) \mid (A \otimes A)$$

Formulae in the form $a, \overline{a}$ are called atomic. We denote by $\overline{a}$ the *negation* of a. The negation of an arbitrary formula $\overline{A}$ is syntactically defined by the following (De Morgan) rules:

$$\begin{aligned}
\overline{(A \parr B)} &\triangleq \overline{B} \otimes \overline{A} \\
\overline{A \otimes B} &\triangleq \overline{B} \parr \overline{A} \\
\overline{\overline{a}} &\triangleq a \\
\overline{\mathbf{1}} &\triangleq \bot \\
\overline{\bot} &\triangleq \mathbf{1}
\end{aligned}$$

3.1 Equivalence between formulae

As we already remarked, the CoS introduces the notion of structures which are equivalence classes of formulae. Formulae contained in the same equivalence class are considered to be elementary logical equivalence. In a derivation there is always the freedom to choose the most suitable representative of a structure. In this way, it is possible to omit what is considered bureaucracy, and so better highlight the important steps in a derivation. Here we follow a different approach, we do not use structures and work directly on formulae. An abstract motivation for our choice is the idea that working with structures, hence with equivalence classes, it is possible to hide some interesting aspects of the proof theory. A more concrete argument against the use of structures is the fact our cut elimination proof relies on the distinction between formulae belonging to the same equivalence class and cannot be presented in term of structures.

Having decide to work directly on formulae, we need to introduce some rules, not explicitly present in the CoS, allowing to substitute, in a derivation, a formulae with by an elementary equivalent one. This can be obtained by introducing a set of rules, each rule stating a particular property of a particular connective. However, in order to have a more compact presentation, we group together

these "equivalence" rules in a single rule. To this end we introduce a relation $\sim$ between formulae; $\sim$ related formulae that can be shown equivalent by a *single* application of the commutativity, associativity and identity laws for the connectives par and tensor. Note that in defining the relation $\sim$ we do not use any symmetric or transitive closure, hence $\sim$ is *not* a equivalence relation. The relation $\sim$ is defined by the following set of schemata:

$$\begin{array}{rcll}
A \parr B & \sim & B \parr A & \text{Par commutative} \\
A \otimes B & \sim & B \otimes A & \text{Times commutative} \\
A \parr (B \parr C) & \sim & (A \parr B) \parr C & \text{Par associative} \\
A \otimes (B \otimes C) & \sim & (A \otimes B) \otimes C & \text{Times associative} \\
 & & & \\
\bot \parr A & \sim & A & \text{Par unit L} \\
A & \sim & A \parr \bot & \text{Par unit R} \\
\mathbf{1} \otimes A & \sim & A & \text{Times unit L} \\
A & \sim & A \otimes \mathbf{1} & \text{Times unit R}
\end{array}$$

3.2 Proof system

We take full advantage, in presenting our calculus, from the fact that logical rules are closed by positive contexts. Positive contexts are generated by the grammar:

$$S \ ::= \ \circ \ \mid \ (A \parr S) \ \mid \ (S \parr A) \ \mid \ (A \otimes S) \ \mid \ (S \otimes A)$$

we denote by $S[A]$ the formula obtained by replacing, in the structural context S, the place holder $\circ$ by the formula A.

The proof system, for multiplicative linear logic, is given by the following set of inference rules:

$$\frac{}{\mathbf{1}} \quad \text{Empty (Emp)}$$

$$\frac{S[B]}{S[A]} \quad \text{if } A \sim B \qquad \text{Equivalence (Eq)}$$

$$\frac{S[\mathbf{1}]}{S[A \parr \overline{A}]} \quad \text{Interaction (Int)}$$

$$\frac{S[A \otimes \overline{A}]}{S[\bot]} \quad \text{Cut}$$

$$\frac{S[(A \parr B) \otimes C]}{S[A \parr (B \otimes C)]} \quad \text{Switch (Sw)}$$

As we already remark, the Equivalence rule can be seen as a compact way to represent a set of inference rules, namely the 8 rules obtain by considering, one by one, the 8 schemata defining the relation $\sim$.

Definition 1. *We denote with* $\vdash A$ *the derivability of the formula* A *by the above defined set of rules.*

The article [11] contains a presentation of multiplicative exponential linear logic (MELL) in the calculus of structures. Apart from the differences in the syntax and in the equivalence rule, we use the same rules presented in [11] for the multiplicative connectives. Similarly to what has been done in [11] we can prove that our calculus satisfies the cut-elimination property and is equivalent to multiplicative linear logic.

Proposition 1. *For any formula* A*,*

(i) $\vdash A$ *if and only if* A *is provable in* MLL*;*
(ii) *if* $\vdash A$ *then* A *is provable without using the Cut rule.*

We omit proofs since they are already present in [11] and can be easily derived by the corresponding proofs for cyclic linear logic given in the next sections.

4 Multiplicative Cyclic Linear Logic

Cyclic Linear Logic can be obtained by simply removing the commutative rules from MLL, that is we consider a new relation $\sim_{\mathsf{N}}$ that is equal to the relation $\sim$, given in Section 3.1, except for the omission of the commutative rules for the par and tensors. However, to have a coherent proof system, we need to substitute the equivalence rule with the a new one having the following form:

$$\frac{S[B]}{S[A]} \quad \text{if } A \sim_{\mathsf{N}} \text{ or } B \sim_{\mathsf{N}} A \qquad \text{EquivalenceN (EqN)}$$

In the commutative calculus, this reflexive formulation of the equivalence rule is not necessary. In fact, through commutativity, it is possible to derive each extra case given by the EquivalenceN rule by (at most three) consecutive applications of the Equivalence rule. Similarly to the Equivalence rule, the EquivalenceN rules can be seen as a compact way to represent a set of inference rules, namely the 12 rules obtain by considering, in the both directions, the 6 schemata defining the relation $\sim_{\mathsf{N}}$.

We need to add also a mirror image version of the Switch rule:

$$\frac{S[A \otimes (B \parr C)]}{S[(A \otimes B) \parr C]} \quad \text{Switch Mirror (SwM)}$$

which is not present in the MLL since is there derivable through commutativity.

Definition 2. *We call system* NLS *the inference calculus formed by the rules: Empty, EquivalenceN, Switch, Switch Mirror, Interaction, and Cut. We denote with* $\vdash_{\mathsf{N}} A$ *the fact that the formula* A *is derivable in system* NLS*.*

Our proof system is equivalent to multiplicative CyLL.

Theorem 1. *For any formula A, $\vdash_{\mathsf{N}} A$ if and only if A is provable in multiplicative* CyLL.

Proof. In order to prove this theorem we consider the presentation of multiplicative CyLL given in [2]. We start by proving the left to right implication, that is, everything provable in our system is provable in multiplicative CyLL. The implication follows immediately from two properties of CyLL. The first one is that derivation is closed by positive context. That is, if S is a positive context not containing negation, and the formulae $S[A]$ and $\overline{A}\parr B$ are derivable, then also the formula $S[B]$ is derivable. This fact can be proved in the following way. Let Π be a derivation, in multiplicative CyLL, for $S[A]$, since Π cannot examine the formula of A until the formula A appears as an element of a sequent, looking at derivations bottom-up-wise, the derivation Π is "independent" from A until it builds a sequent $\Phi(A)$ containing the formula A. From the sequent $\Phi(A)$, using the Cyclic rule it is then possible to derive a sequent Φ', A (having A as last formula) from which, by the Cut rule, one derives Φ', B and, by the Cyclic rule, $\Phi(B)$. From $\Phi(B)$, by following the pattern in Π, one can finally derive $S[B]$.

The second property is that for any rule

$$\frac{S[A]}{S[B]}$$

contained in system NLS, the formula $\overline{A}\parr B$ is derivable in multiplicative CyLL. This fact can be checked straightforwardly.

The other implication is also simple. First, we define a translation, $\_{}_S$, from sequents and sets of sequents, into formulae:

$$\begin{aligned} \underline{A_1, \ldots, A_m}_{\mathsf{S}} &\triangleq \underline{A_1}_{\mathsf{S}} \parr \ldots \parr \underline{A_m}_{\mathsf{S}} \\ \underline{\{\Gamma_1, \ldots, \Gamma_n\}}_{\mathsf{S}} &\triangleq \underline{\Gamma_1}_{\mathsf{S}} \otimes \ldots \otimes \underline{\Gamma_n}_{\mathsf{S}} \end{aligned}$$

It is then easy to check that, any CyLL rule different from the Cycling rule is derivable, that is for any rule in the form:

$$\frac{\vdash \Gamma_1 \ldots \vdash \Gamma_n}{\vdash \Delta} \quad ,$$

from the formula $\underline{\{\Gamma_1, \ldots, \Gamma_n\}}_{\mathsf{S}}$, using the rules in NLS, it is possible to derive the formula $\underline{\Delta}_{\mathsf{S}}$.

Finally, we prove that the Cycling rule is admissible. The Cycling rule has form

$$\frac{\Delta, \Gamma}{\Gamma, \Delta} \text{ Cycling}$$

Its admissibility in system NLS can be express in the following way: if $\vdash A \parr B$ then $\vdash B \parr A$. The proof works as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\mathbf{1}}\,\text{Empty}}{B \parr \overline{B}}\,\text{Interaction}}{B \parr (\mathbf{1} \otimes \overline{B})}\,\text{EquivalenceN}}{\vdots\,\text{Hypothesis}}}{\dfrac{\dfrac{\dfrac{B \parr ((A \parr B) \otimes \overline{B})}{B \parr (A \parr (B \otimes \overline{B}))}\,\text{Switch}}{B \parr (A \parr \bot)}\,\text{Cut}}{B \parr A}\,\text{EquivalenceN}}$$

□

4.1 Cut elimination

A fundamental feature of every logical system is the cut-elimination property, which can also be proved for NLS. If we consider the different logics so far presented in the CoS, [9, 7, 8, 10] and we compare, for these logics, the proofs of cut-elimination in the sequent calculus, and in the CoS, normally we have that the latter proofs are lengthier. This fact can be explained by remarking that, in its complete formulation, the CoS gives more freedom in constructing derivations. It follows that derivations can be quite an anarchic object, and the standard proof technique of structural induction on the complexity of derivations is more difficult to use. A standard technique, for proving cut elimination in the CoS, is to use of the so-called splitting lemma [7]. The splitting lemma states that one can consider just derivations of a particular shape, i.e. a particular subset of derivations is sufficient to derive any provable judgment. We think that the splitting lemma is applicable also to this case, however here we prefer to use a different proof technique. There are two reasons for this choice. The first one is that when we conceive our proof, the splitting lemma was not discovered yet. The second reason is that our proof enlighten some interesting aspect of the CoS, namely the admissibility of some instances of the Equivalence rules. The main idea in our proof is to use formulae instead of structures. Once this choice has been made, the cut elimination proof is obtained by standard techniques. In more detail, we define a restricted calculus with a minimal set of derivations rules. We then prove that all the omitted rules are admissible in the minimal calculus. As immediate consequence, since the cut rule is an omitted one, we have a proof cut elimination. Since we need to show the admissibility of the omitted rules by taking each rule at a time our proof of is quite lengthy also if the single steps, and the general structure, are quite simple.

As a first step, we need to present a restricted calculus having a minimal set of rules. To motivate this restricted calculus we need to introduce the concept of duality between rules. Given of a rule S

$$\frac{A}{B}\ \text{S}$$

the dual of S is the rule

$$\frac{\overline{B}}{\overline{A}}\ \text{dS}$$

For example the rule Interaction is dual to the rule Cut and Switch, Switch Mirror are dual to themselves. It is also easy to observe that from any rule S it is possible to derive its dual, using the Interaction, the Switch, and the Cut rules. In the following, we are going to prove that for any pair of dual rules one of them can be eliminated. Duality between rules is a standard concept in the CoS, and it also standard result the fact that for each each pair of dual rules one of them can be eliminated. What makes our approach different from the previous ones is the fact that we have an explicit rule for equivalence and that we apply the notion of duality also to it. In particular, as we already remark, EquivalenceN rule is a compact way of expressing a set of rules: a rule saying that par is associative, another saying that par is commutative etc. Considering this underlying set of rules, one can observe that it contains pairs of dual rules. For example the rule stating associativity of par is dual to the rule stating associativity of times, the rule for the introduction of the times unit is dual to the rule for the elimination of the par unit, and so on. In the restricted calculus, we insert just one single instance for each pair of dual rules. In doing so we depart from the main stream of the CoS; not only we make the application of the equivalence rule explicit but in the restricted system we do not allow the application of some equivalences. In particular we show that the rules stating the associativity of times are admissible. With this aim, we define a restricted version of the Equivalence rule. This restricted version of the Equivalence considers a new relation, $\rightsquigarrow$, on formulae.

Definition 3. *The relation on formulae $\rightsquigarrow$ is defined as follows:*

$$\begin{array}{rcll}
(A \parr B) \parr C & \rightsquigarrow & A \parr (B \parr C) & \textit{Par associative L} \\
A \parr (B \parr C) & \rightsquigarrow & (A \parr B) \parr C & \textit{Par associative R} \\
\bot \parr A & \rightsquigarrow & A & \textit{Par unit L} \\
A \parr \bot & \rightsquigarrow & A & \textit{Par unit R} \\
\mathbf{1} \otimes A & \rightsquigarrow & A & \textit{Times unit L} \\
A \otimes \mathbf{1} & \rightsquigarrow & A & \textit{Times unit R}
\end{array}$$

The Restricted Equivalence rules is:

$$\frac{S[A]}{S[B]}\ \textit{if } B \rightsquigarrow A \qquad \textit{Restricted Equivalence (REq)}$$

Moreover, it is useful to consider a restricted version of the Interaction rule. In fact, it is possible to reduce the Interaction rule to its atomic version. We call Atomic Interaction the Interaction rule restricted to atomic formulae.

$$\frac{S[\mathbf{1}]}{S[A\parr\overline{A}]} \text{ with } A \text{ atomic formula} \qquad \text{Atomic Interaction}$$

Definition 4. *We call system* $\mathsf{NLS_r}$ *the inference calculus formed by the rules: Empty, Restricted Equivalence, Atomic Interaction, Switch, and Switch Mirror. We write* $\vdash_{\mathsf{r}} A$ *to indicate that the formula* A *is provable in system* $\mathsf{NLS_r}$.

We aim to prove that system $\mathsf{NLS_r}$ is equivalent to the NLS. In particular, we will prove that all the rules in NLS are admissible in $\mathsf{NLS_r}$. The proof proceeds by several steps, each step proving the admissibility of one missing rule.

Lemma 1. *The Interaction rule is derivable in* $\mathsf{NLS_r}$*, that is, for any formula* A*, and context* S*, from* $S[\mathbf{1}]$*, using the* $\mathsf{NLS_r}$ *rules it is possible to derive* $S[A\parr\overline{A}]$.

Proof. By induction on the complexity of the formula A. If A is a unit, then the thesis follows from the Restricted Equivalence rule. In the case where A is an atom, the thesis follows from the Atomic Interaction rule. In the case where A is in the form $A' \otimes A''$ we have the following chain of implications:
$\vdash_{\mathsf{r}} S[\mathbf{1}] \;\Rightarrow$ (by inductive hypothesis)
$\vdash_{\mathsf{r}} S[A'\parr\overline{A'}] \;\Rightarrow$ (by Restricted Equivalence rule)
$\vdash_{\mathsf{r}} S[(A' \otimes \mathbf{1})\parr\overline{A'}] \;\Rightarrow$ (by inductive hypothesis)
$\vdash_{\mathsf{r}} S[(A' \otimes (A''\parr\overline{A''}))\parr\overline{A'}] \;\Rightarrow$ (by Switch Mirror rule)
$\vdash_{\mathsf{r}} S[((A' \otimes A'')\parr\overline{A''})\parr\overline{A'}] \;\Rightarrow$ (by Restricted Equivalence rule)
$\vdash_{\mathsf{r}} S[(A' \otimes A'')\parr(\overline{A''}\parr\overline{A'})]$.
The case where A is in the form $A'\parr A''$, is perfectly equivalent (mirror image) to the previous one. □

Next, we prove admissibility of the Equivalence rule. The proof will be done by induction on the structures of the derivation. To make the induction working, we need to take a stronger, and more involved, inductive hypothesis. A preliminary definition and a lemma are here necessary. We start by defining new classes of contexts.

Definition 5. *(i) A* left context*,* T_l*, is defined by the following grammar:*

$$T_l ::= \circ \;\mid\; A\parr T_l \mid A \otimes T_l$$

symmetrically, a right context*,* T_r *is defined by the grammar:*

$$T_r ::= \circ \;\mid\; T_r\parr A \mid T_r \otimes A$$

(ii) A left par-context*,* V_l*, is defined by the following grammar:*

$$V_l ::= \circ \;\mid\; A\parr V_l$$

symmetrically, a right par-context*,* V_r*, is defined by the grammar:*

$$V_r ::= \circ \;\mid\; V_r\parr A$$

Lemma 2. *For any formulae A, B, C, context S, left par-context V_l and right par-context V_r the following points hold:*

(i) *if* $\vdash_{\mathsf{r}} S[\,V_l[A \otimes B] \otimes C\,]$ *then* $\vdash_{\mathsf{r}} S[\,V_l[A \otimes (B \otimes C)]\,]$, *and symmetrically if* $\vdash_{\mathsf{r}} S[\,A \otimes V_r[B \otimes C]\,]$ *then* $\vdash_{\mathsf{r}} S[\,V_r[(A \otimes B) \otimes C]\,]$,

(ii) *if* $\vdash_{\mathsf{r}} S[\,A \otimes V_r[\mathbf{1}]\,]$ *then* $\vdash_{\mathsf{r}} S[\,V_r[A]\,]$, *and symmetrically, if* $\vdash_{\mathsf{r}} S[\,V_l[\mathbf{1}] \otimes A\,]$ *then* $\vdash_{\mathsf{r}} S[\,V_l[A]\,]$,

(iii) *if* $\vdash_{\mathsf{r}} S[\,\bot \parr A\,]$ *then* $\vdash_{\mathsf{r}} S[A]$ *and symmetrically, if* $\vdash_{\mathsf{r}} S[\,A \parr \bot\,]$ *then* $\vdash_{\mathsf{r}} S[A]$

Proof. All three points are proved by structural induction of the derivation Δ of the judgment in the premise. Here we present just the proof of point (i) which is the most involved one, having the larger number of cases to consider. The other points can be treated with perfectly similar arguments. For each point, the cases to consider concern the last rule, R, applied in the derivation Δ. The simple cases are the ones where R modifies the derived formula only inside one of the contexts S, V_l, V_r, or inside one of the formulae A, B, C: in these cases the thesis simply derives by inductive hypothesis and by an application of R.

For the remaining cases, we schematically present each case by a pair of derivations, the one on left is the application of R considered, while the one on the right shows how the case can be treated, i.e. by, in case, applying the inductive hypothesis, and by the given derivation.

(i.a) The rule R generates one of the formulae A, B, C. This can only happen if one of the formulae is a times unit. The case where A is generated is described and treated as follows:

$$\frac{S[V_l[B] \otimes C]}{S[V_l[\mathbf{1} \otimes B] \otimes C]}\ \mathrm{REq} \qquad\qquad \frac{\begin{array}{c} S[V_l[B] \otimes C] \\ \vdots\ \ (\mathrm{Sw})^* \\ S[V_l[B \otimes C]] \end{array}}{S[V_l[\mathbf{1} \otimes (B \otimes C)]]}\ \mathrm{REq}$$

The cases where B or C are generated can be dealt in a similar way.

(i.b) The last rule R is a Switch rule involving the context (V_l) and the formula $(A \otimes B)$. This interaction can occur in two forms, the first one is the case where $V_l \equiv V_l'[D \parr \circ\,]$, and:

$$\frac{S[V_l'[(D \parr A) \otimes B] \otimes C]}{S[V_l'[D \parr (A \otimes B)] \otimes C]}\ \mathrm{Sw} \qquad\qquad \frac{S[V_l'[(D \parr A) \otimes (B \otimes C)]]}{S[V_l'[D \parr (A \otimes (B \otimes C))]]}\ \mathrm{Sw}$$

(i.c) A second form of interaction between the context (V_l) and the formula $(A \otimes B)$, is given by case where $V_l \equiv V_l'[(D \otimes E) \parr \circ]$, and:

$$\frac{S[V_l'[D \otimes (E \parr (A \otimes B))] \otimes C]}{S[V_l'[(D \otimes E) \parr (A \otimes B)] \otimes C]}\ \mathrm{SwM} \qquad \frac{\dfrac{S(V_l'[D \otimes ((E \parr (A \otimes B)) \otimes C)]]}{S(V_l'[D \otimes (E \parr ((A \otimes B) \otimes C))]]}\ \mathrm{SwM}}{S[V_l'[(D \otimes E) \parr ((A \otimes B) \otimes C)]]}\ \mathrm{Sw}$$

(i.d) The context S interacts with the formula C, in this case $S \equiv S'[\circ \parr D]$ and:

$$\frac{S'[V_l[A \otimes B] \otimes (C \parr D)]}{S'[(V_l[A \otimes B] \otimes C) \parr D]}\ \text{SwM} \qquad \begin{array}{c} S'[V_l[A \otimes (B \otimes (C \parr D))]] \\ \vdots\ \ \text{SwM} \cdot \text{SwM} \\ S'[V_l[(A \otimes (B \otimes C)) \parr D]] \\ \vdots\ \ (\text{REq})^* \\ S'[V_l[A \otimes (B \otimes C)] \parr D] \end{array}$$

(i.e) The context S interacts with the context V_l, in this case $S \equiv S'[D \parr \circ]$ and the last inference rule is:

$$\frac{S'[(D \parr V_l[A \otimes B]) \otimes C]}{S'[D \parr (V_l[A \otimes B] \otimes C)]}\ \text{SwM}$$

for this case, it is sufficient to simply apply the inductive hypothesis.

(i.f) Finally we need to consider the interaction between the context S and the formula $V_l[A \otimes B] \otimes C$, in this case $S \equiv S'[\circ \parr (D \otimes E)]$ and:

$$\frac{S'[((V_l[A \otimes B] \otimes C) \parr D) \otimes E))]}{S'[(V_l[A \otimes B] \otimes C) \parr (D \otimes E)]}\ \text{SwM} \qquad \frac{S'[(V_l[A \otimes (B \otimes C)] \parr D) \otimes E]}{S'[(V_l[A \otimes (B \otimes C)] \parr (D \otimes E)]}\ \text{SwM}$$

The case where $S \equiv S'[(D \otimes E) \parr \circ]$ is equally easy. □

Notice that as a special case of point (i) and (ii) of the above lemma we have that

(i) if $\vdash_{\mathsf{r}} S[(A \otimes B) \otimes C]$ then $\vdash_{\mathsf{r}} S[A \otimes (B \otimes C)]$, and symmetrically if $\vdash_{\mathsf{r}} S[A \otimes (B \otimes C)]$ then $\vdash_{\mathsf{r}} S[(A \otimes B) \otimes C]$,
(ii) if $\vdash_{\mathsf{r}} S[A \otimes \mathbf{1}]$ then $\vdash_{\mathsf{r}} S[A]$, and symmetrically, if $\vdash_{\mathsf{r}} S[\mathbf{1} \otimes A]$ then $\vdash_{\mathsf{r}} S[A]$,

it follows:

Proposition 2. *The EquivalenceN rule is admissible in* $\mathsf{NLS}_{\mathsf{r}}$.

Next we proof the admissibility of the atomic cuts, also for this case the proof is done by induction on the structures of the derivation, and also for this case to make the induction working, we need to take a stronger, and more involved, inductive hypothesis. The following lemma implies the admissibility of atomic cuts.

Lemma 3. *For any atom a, context S, left context T_l and right context T_r, if* $\vdash_{\mathsf{r}} S[T_l[a] \otimes T_r[\overline{a}]]$ *(or* $\vdash_{\mathsf{r}} S[T_l[\overline{a}] \otimes T_r[a]]$*) then* $\vdash_{\mathsf{r}} S[T_l[\bot] \parr T_r[\bot]]$.

Proof. The proof is by structural induction on the derivation Δ of $\vdash_{\mathsf{r}} S[T_l[a] \otimes T_r[\overline{a}]]$, and it is quite similar to the proof of Lemma 2. The different cases considered by the structural induction can be split in two groups. The simple cases are the ones where the last rule R in Δ works internally to one of the contexts S, T_l, T_r; in all these cases the thesis follows by inductive hypothesis and by an application of the rule R. The other cases are the ones where the rule R modifies more than one context, or generates one of the atoms a, $\overline{a}$. In detail:

(a) an Atomic Interaction rule generates one of the atoms. This case can be described and treated as follows:

$$\frac{S[T_l[a] \otimes T'_r[\mathbf{1}]]}{S[T_l[a] \otimes T'_r[\overline{a} \parr a]]}\ \text{AInt} \qquad \begin{array}{c} \dfrac{S[T_l[a] \otimes T'_r[\mathbf{1}]]}{S[T_l[\bot \parr a] \otimes T'_r[\mathbf{1}]]}\ \text{REq} \\ \vdots\ (\text{REq} + \text{SwM})^* \\ \dfrac{S[(T_l[\bot] \parr a) \otimes T'_r[\mathbf{1}]]}{S[T_l[\bot] \parr (a \otimes T'_r[\mathbf{1}])]}\ \text{Sw} \\ \vdots\ (\text{EqN} + \text{SwM})^* \\ \dfrac{S[T_l[\bot] \parr T'_r[(a \otimes \mathbf{1})]]}{S[T_l[\bot] \parr T'_r[a]]}\ \text{EqN} \\ \dfrac{S[T_l[\bot] \parr T'_r[a]]}{S[T_l[\bot] \parr T'_r[\bot \parr a]]}\ \text{REq} \end{array}$$

One should remark that the one of the right is not a true derivation in $\mathsf{NLS}_{\mathsf{r}}$, in fact the EquivalenceN rule is just admissible in $\mathsf{NLS}_{\mathsf{r}}$. The right diagram should be interpreted as a schematic proof that the formula $S[T_l[\bot] \parr T'_r[\bot \parr a]]$ is derivable in $\mathsf{NLS}_{\mathsf{r}}$.

(b) a Switch rule makes the contexts S and T interact. This case can be described and treated as follows: $S \equiv S'[A \parr \circ]$ and

$$\frac{S'[(A \parr T_l[a]) \otimes T_r[\overline{a}]]}{S'[A \parr (T_l[a] \otimes T_r[\overline{a}])]}\ \text{Sw} \qquad \frac{S'[(A \parr T_l[\bot]) \parr T_r[\bot]]}{S'[A \parr (T_l[\bot] \parr T_r[\bot])]}\ \text{REq}$$

It remains to consider the cases where the context S interacts with the whole formula $T_l[a] \otimes T_r[\overline{a}]$, where the contexts T_l interact with formula a and where the contexts T_r interact with formula $\overline{a}$. All these cases are immediate. □

Lemma 4. *The Cut rule is admissible in* $\mathsf{NLS}_{\mathsf{r}}$*, that is, for any formula* A*, context* S*, if* $\vdash_{\mathsf{r}} S[A \otimes \overline{A}]$ *then* $\vdash_{\mathsf{r}} S[\bot]$.

Proof. This lemma can be seen as a sort of dual of Lemma 1 By structural induction on the formula A. If A is a unit then the thesis follows from the admissibility of the EquivalenceN rule.

In the case where A is an atom, the thesis follows from the previous lemma and from the admissibility of the EquivalenceN rule.

The case where A is in the form $A' \parr A''$ can be treated as follows:

$$\dfrac{\dfrac{\dfrac{S[(A' \parr A'') \otimes (\overline{A''} \otimes \overline{A'})]}{S[((A' \parr A'') \otimes \overline{A''}) \otimes \overline{A'}]}\ \text{EqN}}{S[(A' \parr (A'' \otimes \overline{A''})) \otimes \overline{A'}]}\ \text{Sw}}{\vdots\ \text{Inductive hypothesis}}$$

$$\dfrac{\dfrac{S[(A' \parr \bot) \otimes \overline{A'}]}{S[A' \otimes \overline{A'}]}\ \text{EqN}}{\vdots\ \text{Inductive hypothesis}}$$

$$S[\bot]$$

Note that the above is not a true derivation but just a schematic proof of the derivability of $S[\bot]$. The case where A is in the form $A' \otimes A''$ is perfectly equivalent to this one. □

Having proved that all the rules in NLS are admissible in $\mathsf{NLS_r}$ we can finally state:

Proposition 3. *For every formula A if $\vdash_{\mathsf{N}} A$ the $\vdash_{\mathsf{r}} A$.*

That is, the restricted system $\mathsf{NLS_r}$ is as powerful as the complete one NLS and from this we have:

Theorem 2. *The system NLS satisfies the cut-elimination property.*

5 Further Works

A natural question to consider is whether the above treatment for the multiplicative cycling logic can be extended to richer logics. In particular, one should consider the complete system of the cycling linear logic [2] and the multiplicative non-commutative logic of Abrusci and Ruet [5]. The complete cyclic linear logic extend the multiplicative part, considered here, by adding the missing linear logic connectives. While the multiplicative non-commutative logic contains both commutative and non-commutative multiplicative connectives.

Without giving any proof we claim that the first extension can be carried out quite smoothly; it is sufficient to consider the presentation of LL in the CoS given in [10] and modify it by removing the commutativity rules. In this way we obtain a proof system for cyclic linear logic. In this system cut elimination can be proved using the technique presented in this article, with the only extra difficulty of using a more complex induction to deal with the exponential and additive connectives. In fact these connectives can multiply the occurrences of a formula in the premises.

The treatment of the multiplicative non-commutative linear logic [5] is still an open problem. We remark that it is possible to formulate, in the CoS, all the rules of the non-commutative logic, given, in the sequent calculus formulation, in [6]. However this presentation will be an obvious and uninteresting result. In this way deep inference will not play any role. A more interesting application of the CoS in this setting would be a proof that the Seesaw rule (the non commutative logic equivalent to the Cycling rule) can be substituted by deep inference. However so far we were not able to find a nice formulation for the non-commutative logic, in the CoS. Our difficulties can be explained by the fact that deep inference alone, in the non-commutative calculus, is not able to reduce the Interaction rule to the Atomic Interaction rule, and dually Cut to and atomic form of cut. These reductions are possible instead in the other logics and are a key ingredient in the cut elimination proofs.

Acknowledgments

We thank the anonymous referees for their useful suggestions and Alessio Guglielmi for many useful suggestions and stimulating discussions.

References

1. Girard, J.Y.: Linear logic. Theoretical Computer Science **50** (1987) 1–102
2. Yetter, D.N.: Quantales and (non-commutative) linear logic. J. Symbolic Logic **55** (1990)
3. Mulvey, C.J.: &. In: Second topology conference. Number 12 in Rediconti del Circolo Matematico di Palermo (1986) 94–104
4. Lambek, J.: The mathematics of sentece structures. Amer. Math. Mon. **65** (1958) 154–170
5. Abrusci, V.M., Ruet, P.: Non-commutative logic I: the multiplicative fragment. Annals of Pure and Applied Logic **101** (2000) 29–64
6. Ruet, P.: Non-commutative logic II: sequent calculus and phase semantics. Mathematical Structures in Computer Science **10** (2000) 277–312
7. Guglielmi, A.: A system of interaction and structure. Technical Report WV-02-10, Dresden University of Technology (2002) Conditionally accepted by ACM Transactions on Computational Logic, http://pikas.inf.tu-dresden.de/~guglielm/paper.html.
8. Stewart, C., Stouppa, P.: A systematic proof theory for several modal logics. Technical Report WV-03-08, Technische Universität Dresden (2003)
9. Brunnler, K.: Atomic cut elimination for classical logic. In Baaz, M., Makowsky, J.A., eds.: CSL 2003. Volume 2803 of Lecture Notes in Computer Science., Springer-Verlag (2003) 86–97
10. Straßburger, L.: A local system for linear logic. In: LPAR 2002. Volume 2514 of Lectures Notes in Artificial Intelligence., Springer-Verlag (2002) 388–402
11. Guglielmi, A., Straßburger, L.: Non-commutativity and MELL in the calculus of structures. In Fribourg, L., ed.: CSL 2001. Volume 2142 of Lecture Notes in Computer Science., Springer-Verlag (2001) 54–68