Focusing and Proof-Nets in Linear and Non-commutative Logic

Jean-Marc Andreoli[1] and Roberto Maieli[2*]

[1] Xerox Research Centre Europe, 38240 Meylan, France
Jean-Marc.Andreoli@xrce.xerox.com,
Web: http://www.xrce.xerox.com/people/andreoli
[2] Logica - Università "Roma 3", 00154 Roma, Italy
maieli@phil.uniroma3.it

Abstract. Linear Logic [4] has raised a lot of interest in computer research, especially because of its resource sensitive nature. One line of research studies proof construction procedures and their interpretation as computational models, in the "Logic Programming" tradition. An efficient proof search procedure, based on a proof normalization result called "Focusing", has been described in [2]. Focusing is described in terms of the sequent system of commutative Linear Logic, which it refines in two steps. It is shown here that Focusing can also be interpreted in the proof-net formalism, where it appears, at least in the multiplicative fragment, to be a simple refinement of the "Splitting lemma" for proof-nets. This change of perspective allows to generalize the Focusing result to (the multiplicative fragment of) any logic where the "Splitting lemma" holds. This is, in particular, the case of the Non-Commutative logic of [1], and all the computational exploitation of Focusing which has been performed in the commutative case can thus be revised and adapted to the non commutative case.

1 Introduction

Linear Logic [4] has raised a lot of interest in computer research, especially because of its resource sensitive nature. One line of research, supported by systems such as LO [3], Lambda-Prolog [8], Forum [9] or Lolli [7], studies proof construction procedures and their interpretation as computational models, in the "Logic Programming" tradition. An efficient proof-search procedure for Linear Logic, based on a proof normalization result called "Focusing", has been described in [2]. Focusing is described there in terms of the sequent system of (commutative) Linear Logic, which it refines in two steps ("Dyadic", resp. "Triadic" system). Basically, each refinement eliminates redundancies in proof-search due

[*] This work was performed while the second author was visiting XRCE; this visit was supported by the European TMR (*Training and Mobility for Researchers*) Network "Linear Logic in Computer Science" (esp. the Rome and Marseille sites, XRCE being attached to the latter).

to irrelevant sequentializations of inference figures in the sequent-based representation of proofs. The expressive power of Focusing is captured in a crisp way in a fully representative fragment of Linear Logic, called "LinLog", introduced in [2] together with a normalization procedure from Linear Logic to LinLog. This procedure allows to represent in LinLog all the fragments considered in the various systems mentioned above.

It is shown here that Focusing can also be interpreted in the proof-net formalism, where it appears, at least in the multiplicative fragment, to be a simple refinement of the "Splitting lemma" for proof-nets. The Splitting lemma is at the core of the Sequentialization procedures for proof-nets, and Focusing thus appears as a sequentialization strategy. This change of perspective allows to generalize the Focusing result to (the multiplicative fragment of) any logic where the "Splitting lemma" holds. This is, in particular, the case of the Non-Commutative logic of [1], and all the computational exploitation of Focusing which has been performed in the commutative case can thus be revised and adapted to the non commutative case. The expected outcome of such a program is a finer model of computational resources and agent-based coordination of these resources.

But beyond the technical results, the aim of this paper is to show that Focusing is not limited to a technique adapted to the specific problem of computational proof search, although that was its original motivation (in the line of uniform proofs for Intuitionistic Logic [10]). Focusing is an intrinsic property of resource-conscious logics which admit an involutive duality. It captures in a single framework the quite straightforward and well-known property of "invertibility" of some connectives (called "asynchronous" or negative) together with the not-so-well-known dual of this property which applies to the dual connectives (called "synchronous" or positive), through so-called "critical focusing sections". Focusing, just as Cut-elimination, is a purely logical property, and it is not surprising that it appears under different forms in different contexts, for instance in sequent systems (through search procedures), or in proof-nets (through sequentialization), or even in the more ambitious program of reformulation of Logic known as "Ludics" [5, 6].

Section 2 recalls prior art and notations exploited in this paper. Section 3 describes the main result of this paper, i.e. a reformulation of Focusing in terms of proof-nets and its application to Non-commutative logic.

2 Notations and Prior Art

2.1 Notations

We consider here the multiplicative fragment of Linear Logic (resp. Non-commutative Logic). The connectives are split into two categories:

- Asynchronous: $\parr$ (par), and, in the Non-Commutative case, ∇ (sequential)
- Synchronous: $\otimes$ (times), and, in the Non-Commutative case, $\odot$ (next)

Formulae are built from a given class of atomic formulae using the above connective. A non atomic formula is said to be asynchronous (resp. synchronous) if its top-most connective is asynchronous (resp. synchronous). We assume an involutive duality operation on atomic formulae, generalized to all the formulae using the traditional De Morgan laws:

$$\begin{array}{ll} (A \parr B)^{\perp} = B^{\perp} \otimes A^{\perp} & (A \otimes B)^{\perp} = B^{\perp} \parr A^{\perp} \\ (A \nabla B)^{\perp} = B^{\perp} \odot A^{\perp} & (A \odot B)^{\perp} = B^{\perp} \nabla A^{\perp} \end{array}$$

Furthermore, we assume that the class of atomic formulae is split into two dual, disjoint subclasses, called the positive (resp. negative) atoms.

2.2 Sequent Proofs and Focusing

- Identity rules

$$[\mathrm{I}]\ \frac{}{\vdash F, F^{\perp}} \qquad [\mathrm{C}]\ \frac{\vdash \Gamma, F \quad \vdash \Delta, F^{\perp}}{\vdash \Gamma, \Delta}$$

- Logical rules

$$[\parr]\ \frac{\vdash \Gamma, F, G}{\vdash \Gamma, F \parr G} \qquad [\otimes]\ \frac{\vdash \Gamma, F \quad \vdash \Delta, G}{\vdash \Gamma, \Delta, F \otimes G}$$

Fig. 1. The standard sequent system of Multiplicative Linear Logic

In the fragment of Linear Logic we consider, the standard sequent system is limited to the one shown in Figure 1. Sequents are simple multisets of formulae. Proofs are obtained by assembling in a connected way instances of the inference figures; the assembling is possible when the conclusion of an instance of inference figure is the premiss of another. The resulting structure is a tree labeled with sequents.

Proof search in this system comes up against two snags, identified in [2]: (*i*) two proofs can be equivalent up to some irrelevant permutation of inference figures; (*ii*) two proofs can also be equivalent up to the presence of some "dummy" sub-proofs in which the premisses are all identical and identical to the conclusion (such dummy sub-proofs can simply be discarded). A proof search procedure should not make costly non deterministic choices to distinguish between such pairs of equivalent proofs.

The technique proposed in [2] to deal with these problems relies on a refinement of the sequent system. This refinement satisfies the following main properties:

- Each inference figure in the refined system is a combination of inference figures of the initial one. Hence, each proof in the refined system corresponds straightforwardly to a proof in the initial one. This mapping is called "transduction".

- Each proof in the initial system is equivalent (modulo permutations of inference figures and deletion of dummy sub-proofs, of the kind mentioned above) to a proof obtained by transduction of a proof in the refined system.

In other words, proofs in the refined system fully represent proofs in the initial system, except that the refined system does not distinguish between many equivalent proofs of the initial system, which differ only by irrelevant syntactical differences. Hence, proof search in the refined system yields basically the same proofs and proof constructions as in the initial system, but saves a lot of resources otherwise needed to manage irrelevant non-determinism in the proof search process.

- Logical rules

$$[\parr]\ \frac{\vdash \Gamma \Uparrow L, F, G}{\vdash \Gamma \Uparrow L, F \parr G} \qquad [\otimes]\ \frac{\vdash \Gamma \Downarrow F \quad \vdash \Delta \Downarrow G}{\vdash \Gamma, \Delta \Downarrow F \otimes G}$$

- Reaction $\Uparrow$: if F is not asynchronous

$$[R\Uparrow]\ \frac{\vdash \Gamma, F \Uparrow L}{\vdash \Gamma \Uparrow L, F}$$

- Reaction $\Downarrow$: if F is neither synchronous nor a positive atom

$$[R\Downarrow]\ \frac{\vdash \Gamma \Uparrow F}{\vdash \Gamma \Downarrow F}$$

- Identity: if F is a positive atom

$$[\mathrm{I}]\ \frac{}{\vdash F^{\perp} \Downarrow F}$$

- Decision: if F is synchronous or a positive atom

$$[D]\ \frac{\vdash \Gamma \Downarrow F}{\vdash \Gamma, F \Uparrow}$$

Fig. 2. The Focusing sequent system for Multiplicative Linear Logic

In the fragment of logic we consider, the refined system described in [2] can be reduced to the one shown in Figure 2. It is called below the "Focusing" system. Focusing sequents are of two types:

1. $\Gamma \Uparrow L$ where Γ is a multiset of non-asynchronous formulae and L an ordered list of formulae;
2. $\Gamma \Downarrow F$ where Γ is a multiset of non-asynchronous formulae and F is a single formula (called the "focus").

The transduction of a Focusing inference figure simply "forgets" the structure of the Focusing sequents (i.e. $\Gamma \Uparrow L$ becomes Γ, L where the order in L is forgotten and $\Gamma \Downarrow F$ becomes Γ, F). In the case of the logical and identity inference figures, transduction yields the corresponding inference figure in the initial system. The transduction of the other Focusing inference figures (Reactions and Decision) yields "dummy" inferences in which the premiss is identical to the conclusion (eliminated in the transduction of a proof).

For a discussion of the Focusing system and its computational interpretations in terms of proof search, consult [2]. Notice a slight difference in conventions w.r.t. [2]: here, the Identity rule can only be triggered by a positive atom in the focus (in $\Downarrow$ sequents). In [2], negative atoms had this triggering role, but clearly, polarities are purely conventional, so this difference is only superficial. The Focusing system is justified by the following theorem (stated and proved in [2]):

Theorem 1 (Andreoli 1992). *Let Γ be a multiset of non-asynchronous formulae and L an ordered list of formulae.*

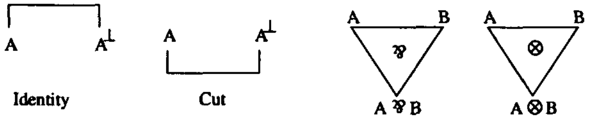
$$\vdash \Gamma, L \text{ if and only if } \vdash \Gamma \Uparrow L$$

More precisely, any proof of Γ, L in the standard sequent system can be mapped, by permutation of inferences and deletion of dummy sub-proofs, into (the transduction of) a proof of $\Gamma \Uparrow L$ in the Focusing system.

There is no straightforward way to map the demonstration of Theorem 1 to the Non-Commutative case. The shape of the focusing sequents in this case is not obvious, and especially it is not clear how to combine the structuring of sequents brought by Focusing with that induced by non-commutativity. Hence the need to consider proof-nets, where the mapping between commutative and non-commutative proofs is more straightforward.

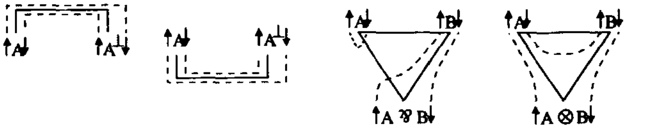
2.3 Proof-Nets and Splitting

Proof-nets have been designed in an attempt to abstract away the inessential sequentializations inherent in the syntax of sequent systems. Proof-nets are defined in two steps. First, proof structures are defined as simple constructions made of nodes and links. Each node is labeled by a single formula. Links are instances of the following prototypes:



In assembling nodes and links in a proof structure, the following purely syntactical conditions must be respected: (*i*) each node is attached to exactly one conclusion of a link and at most one premiss; (*ii*) no two different nodes can be attached to the same premiss or conclusion of a link; (*iii*) the overall structure is connected. The conclusions of a proof structure are the nodes which are not attached to the premiss of any link.

Proof-nets are proof structures which satisfy a certain correctness criterion. Several equivalent criterions have been proposed in the literature. We use here the criterion based on switching positions and paths: each node in a proof-structure is labeled by a formula, but is also decorated by two "gates" (written $\uparrow$ and $\downarrow$); a switching position for a link is an undirected graph between the gates of its premiss and conclusion nodes, of one of the following types (dashed lines):



Right switches (Left switches are symmetric)

The "no short-path" criterion [4] states:

> A proof-net is a proof-structure such that for any choice of a switching position for each of its links, the undirected graph induced between its gates, completed by edges $A \uparrow, A \downarrow$ for each conclusion A, contains a single circuit which goes through all the gates of the proof structure.

Furthermore, we make two technical assumptions, justified by our proof search orientation, and which cost no generality:

- The identity link is restricted to atomic formulae only: any identity link with non atomic formulae can be reduced in a straightforward way to atomic identities.
- The cut link is not used: we make use here of the well known cut-elimination result on proof-nets, proved in [4].

Any sequent proof β can straightforwardly be mapped into a proof structure β^* such that the multiset of conclusions of β^* is exactly the conclusion sequent of β. The equivalence between sequent proofs and proof-nets is precisely given by the following theorem (stated and proved in [4]):

Theorem 2 (Girard 1987). *Equivalence between proof-nets and sequent proofs.*

- *Let β be a sequent proof. Then β^* is a proof-net.*
- *Let π be a proof-net. Then there exists a sequent proof β such that $\beta^* = \pi$.*

The first statement of the theorem is straightforward. The second one relies essentially on the following "Splitting lemma", which we detail here since it is essential to our purpose.

Definition 1. *Let π be a proof-net and F be one of its synchronous conclusions. F is splitting for π, and we write $F \in$ `split`(π) if and only if π consists of two proof nets π_A, π_B plus a synchronous link the premisses of which are conclusions of, resp., π_A and π_B, and the conclusion of which is labeled with F.*

The Splitting lemma (stated and proved in [4]) expresses that, under some conditions, a proof-net can always be split in the sense of the above definition.

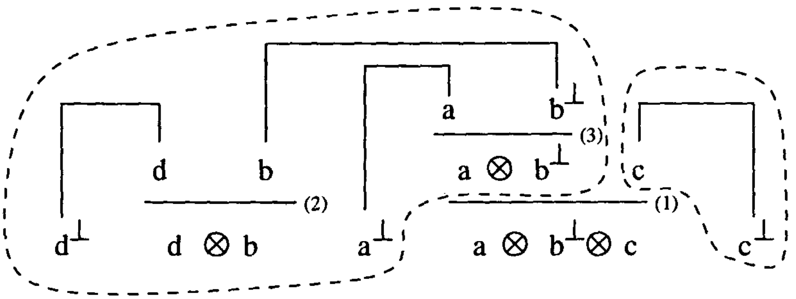


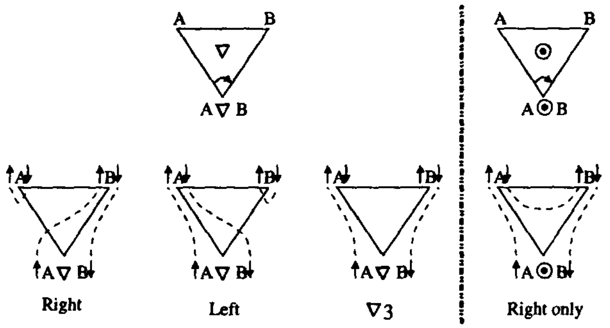
Fig. 3. A sample proof-net and a possible split

Theorem 3. *Let π be a proof-net that contains no asynchronous conclusion and at least one synchronous conclusion. Then* split(π) $\neq \emptyset$

An example of split proof-net is given in Figure 3. The split formula in this case is $a \otimes b^\perp \otimes c$. It is easy to check that the two sub-proof-structures obtained by splitting the net at this conclusion are indeed proof-nets. Notice that there is another splitting conclusion, namely $d \otimes b$.

2.4 The Non-commutative Case

Non-Commutative logic, introduced in [1], is a refinement of the commutative case in terms of proof-nets. Two new link types are added (notice here that the premisses are directed), with associated switching positions:



The criterion for proof-net correctness is extended to the non commutative versions of the connectives. A straightforward mapping between non-commutative proof structures and commutative ones is defined by: given a non commutative proof structure π, we build the corresponding commutative proof structure π^o by replacing in π the occurrences of non-commutative connectives and links by their corresponding commutative version (i.e. $\nabla \mapsto ⅋$ and $\odot \mapsto \otimes$). We then have the following theorem (stated and proved in [1]):

Theorem 4 (Abrusci-Ruet 1998). *Let π be a non-commutative proof structure. π is a (non-commutative) proof-net if and only if*

- *π° is a (commutative) proof-net;*
- *For every $\nabla 3$-free switching for π, the inner parts of ∇-links in the induced cycle contain no conclusions and do not overlap.*

This central theorem allows us to map proof net properties, in particular, as we will see below, Focusing, from the commutative case to the non-commutative one. For a precise definition of "inner parts" of ∇-links and their "overlapping", please refer to [1].

3 Focusing with Proof-Nets

Informally, the main point of this section is to express focusing as a refinement of Theorem 3. This theorem states that whenever a proof-net contains no asynchronous conclusion and at least one synchronous conclusion, there exists a splitting synchronous conclusion. The main refinement we introduce is that the splitting conclusion can be chosen in such a way that each of its premisses, if it is synchronous, is itself a splitting conclusion for the sub-proof-net obtained by splitting. Focusing thus appears as a "hereditary" version of Splitting.

3.1 Focusing Conclusions

In Linear Logic, the sequentialization of a proof-net proceeds by induction on the size of the proof-net. At each induction steps, there are three cases to consider:

- If the proof-net contains an asynchronous conclusion, then
  1. remove the corresponding link;
  2. recursively apply sequentialization to the remaining proof-net;
  3. complete the sequent proof obtained with the corresponding asynchronous inference figure.
- If the proof-net contains no asynchronous conclusion but at least one synchronous conclusion, then
  1. use Theorem 3 to choose a splitting synchronous conclusion, and split the proof-net at this formula into two sub-proof-nets;
  2. recursively apply sequentialization to each of these sub-proof-nets;
  3. combine the resulting sequent proofs with the corresponding synchronous inference figure.
- If the proof-net contains neither synchronous nor asynchronous conclusions, i.e. it must be an instance of the identity link, and its sequentialization is reduced to the identity axiom [I].

This procedure yields a sequent proof the conclusion of which is the sequent made of the multiset of conclusions of the initial proof-net. However, the resulting proof may not be a focusing proof. For example, the sequentialization of the proof-net of Figure 3, where the conclusions are numbered according to the order in which they are chosen for splitting, yields the following sequent proof:

$$[\otimes_1]\,\dfrac{[\otimes_2]\,\dfrac{[\mathrm{I}]\,\dfrac{}{\vdash d^\perp, d} \qquad [\otimes_3]\,\dfrac{[\mathrm{I}]\,\dfrac{}{\vdash b^\perp, b} \quad [\mathrm{I}]\,\dfrac{}{\vdash a, a^\perp}}{\vdash b, a^\perp, a \otimes b^\perp}}{\vdash d^\perp, d \otimes b, a^\perp, a \otimes b^\perp} \qquad [\mathrm{I}]\,\dfrac{}{\vdash c, c^\perp}}{\vdash d^\perp, d \otimes b, a^\perp, a \otimes b^\perp \otimes c, c^\perp}$$

This proof is not focusing (i.e. it cannot be obtained as the transduction of a proof in the Focusing system). Indeed, the inference figure $[\otimes_1]$ decomposes the synchronous formula $a \otimes b^\perp \otimes c$, but its synchronous sub-formula $a \otimes b^\perp$ is not principal in the next inference figure $[\otimes_2]$, violating the "synchronous critical section" property of focusing proofs. To obtain a focused version of the above proof (assuming a, b, c, d are positive atomic), it is here sufficient to permute the inference figures $[\otimes_2]$ and $[\otimes_3]$. Indeed, the proof thus obtained is the transduction of the following proof in the Focusing system:

$$[R{\Uparrow}^*, D]\,\dfrac{[\otimes_1]\,\dfrac{[\otimes_3]\,\dfrac{[\mathrm{I}]\,\dfrac{}{\vdash a^\perp \Downarrow a} \qquad [R{\Downarrow}, D]\,\dfrac{[\otimes_2]\,\dfrac{[\mathrm{I}]\,\dfrac{}{\vdash d^\perp \Downarrow d} \quad [\mathrm{I}]\,\dfrac{}{\vdash b^\perp \Downarrow b}}{\vdash d^\perp, b^\perp \Downarrow d \otimes b}}{\vdash d^\perp, d \otimes b \Downarrow b^\perp}}{\vdash d^\perp, d \otimes b, a^\perp \Downarrow a \otimes b^\perp} \qquad [\mathrm{I}]\,\dfrac{}{\vdash c^\perp \Downarrow c}}{\vdash d^\perp, d \otimes b, a^\perp, c^\perp \Downarrow a \otimes b^\perp \otimes c}}{\vdash\Uparrow d^\perp, d \otimes b, a^\perp, a \otimes b^\perp \otimes c, c^\perp}$$

This focused proof could also have been obtained by sequentialization of the initial proof-net, using a different ordering in the choice of splitting conclusions, namely (1-3-2):

$$a \otimes b^\perp \otimes c \quad , \quad a \otimes b^\perp \quad , \quad d \otimes b$$

instead of (1-2-3):

$$a \otimes b^\perp \otimes c \quad , \quad d \otimes b \quad , \quad a \otimes b^\perp$$

Thus, Focusing basically appears as a strategy in the choice of the splitting formula allowed by Theorem 3 in the Sequentialization procedure. More precisely, Focusing expresses that in the conditions of Theorem 3, not only the set of splitting conclusions is not empty, but its subset, consisting of the "focusing" conclusions, defined below, is also non-empty. Sequentialization will yield a focusing proof if, at each choice of a splitting conclusion in the procedure, a focusing conclusion is selected. The set of focusing conclusions of a net is inductively defined as follows:

Definition 2. *Let π be a proof-net and F be one of its conclusions. F is focusing for π, and we write $F \in \mathtt{foc}(\pi)$ if and only if one of the following two conditions holds:*

1. *F is a positive atom and π is reduced to an axiom link.*
2. *$F \in \mathtt{split}(\pi)$ and π is split at F (with subformulae A and B) into two sub-proof-nets π_A, π_B and*

- *A is asynchronous or a negative atom or $A \in$* $\mathtt{foc}(\pi_A)$
- *B is asynchronous or a negative atom or $B \in$* $\mathtt{foc}(\pi_B)$

From this definition, it is clear that for a proof-net (not reduced to an axiom link), the set of focusing formulae is a subset of the set of splitting formulae.

$$\mathtt{foc}(\pi) \quad \subseteq \quad \mathtt{split}(\pi)$$

Notice however that, unlike splitting, our definition of focusing also applies to proof-nets reduced to the axiom link. This is essential and allows to capture the particular role of polarities in Focusing, which is fully exploited in LinLog, the normalization procedure for Linear Logic [2].

3.2 The Focusing Theorem

The following theorem is shown in Appendix A.2.

Theorem 5. *Let π be a proof-net containing no asynchronous conclusion. Then* $\mathtt{foc}(\pi) \neq \emptyset$.

Thus, Focusing appears as a refinement of Splitting. It expresses a form of "hereditary" Splitting, and, in addition, allows a form of control of the hereditary splitting sequences by the polarities of the atoms found at the end of each sequence (if any). We can now make more precise the view of Focusing as a Splitting strategy in the Sequentialization procedure, illustrated above. For technical reasons, we assume that any proof-net is equipped with a total ordering of its conclusions, which can be straightforwardly expanded to all its nodes in such a way that (*i*) the lowest of two subformulae of the same conclusion is the "left-most, outermost" in the tree representation of that conclusion[1], and (*ii*) the subformulae of different conclusions are in the same order as these conclusions. The ordering of the conclusions can be completely arbitrary; its extension to all the nodes of the proof-net is uniquely defined and induces an ordering of the conclusions for all the sub-proof-nets of the initial one. The ordering is only used here to capture arbitrary choices in the Sequentialization procedure (it has nothing to do with the ordering induced by non-commutativity). Let's enforce that,

- at each choice of an asynchronous conclusion for decomposition in the Sequentialization procedure, the highest (w.r.t. node ordering) asynchronous conclusion is selected;
- at each choice of a synchronous conclusion for decomposition in the Sequentialization procedure (when no asynchronous conclusion exist), the highest (w.r.t. node ordering) *focusing* conclusion is selected.

Then, the following property can easily be shown by induction on the size of the proof-net:

[1] By convention, a formula is "outer" than its own sub-formulae, and in a formula $F\ c\ G$ — where c is any connective — the subformulae of F are "on the left" of those of G.

> Let π be a proof-net and L be the (ordered) list of its conclusions. Then the sequentialization of π is (a transduction of) a Focusing proof of $\vdash\Uparrow L$.

The induction works on this property together with the following one:

> Let π be a proof-net with no asynchronous conclusion and at least one synchronous one (hence at least one focusing conclusion). Then the sequentialization of π is (a transduction of) a Focusing proof of a sequent of the form $\vdash \Gamma \Downarrow F$ where F is the highest *focusing* conclusion of π.

A careful analysis of the proof of Theorem 5 shows that it relies on two basic features: (*i*) the Splitting lemma and (*ii*) the partition of compound formulae between asynchronous and synchronous formulae, completed by the partition of atomic formulae between positive and negative atoms. The Splitting lemma itself has been reformulated in terms of the asynchronous/synchronous duality in Theorem 3. In fact, the proof of Theorem 5 also makes use of an implicit property (a "Merging" property, the proof of which is quite straightforward):

> Let π_A, π_B be two proof-nets. Then the proof structure obtained by assembling π_A, π_B plus a synchronous link the premisses of which are conclusions of, resp., π_A and π_B, is a proof net.

Consequently, Focusing applies to any logic where

- the synchronous/asynchronous duality holds, and
- the (reformulated version of the) Splitting lemma (and Merging property) hold.

This is the case, for instance, of Multiplicative Non-commutative Logic, as shown in Appendix A.3 (for the Splitting lemma) and Appendix A.4 (for Merging) using only Theorem 4. Therefore, Theorem 5 also holds in this Non-commutative Logic. On the other hand, Theorem 5 does not apply to other logics where the Splitting lemma and the asynchronous/synchronous duality do not hold, such as Pomset logic [11] (the connective $<$ is neither synchronous nor asynchronous).

The link between commutative and non-commutative proof-nets, captured by Theorem 4, and the exact analogy of the Focusing property in the commutative and non commutative cases, show that

Theorem 6. *Let π be a non-commutative proof-net.*

$$\mathtt{foc}(\pi^\circ) \;=\; (\mathtt{foc}(\pi))^\circ$$

In particular, this means that, in terms of proof search, the synchronous/asynchronous duality does not distinguish between the commutative and non-commutative cases.

4 Conclusion and Future Work

We have shown here that Focusing can be expressed in terms of proof-nets, when restricted to the multiplicative fragment of Linear Logic. The only property which is used in the demonstration of this result is the "Splitting lemma", reformulated in terms of the "asynchronous/synchronous" duality of the connectives. Consequently, the result can be generalized to any logic where this property holds, and in particular Non-commutative Logic.

But the sequent system version of Focusing, presented in [2] has the interesting property that it applies to whole Linear Logic, not just its multiplicative fragment. Indeed, the deep symmetry captured by the synchronous/asynchronous duality extends straightforwardly to additive connectives, and even, to some extent, to the exponentials, although, in the latter case, the asynchronous behavior of ? and the synchronous behavior of ! appear only in the "dyadic" sequent system, with some adjustments with respect to the other connectives.

As future work, we intend to re-formulate the Focusing result, obtained here in terms of multiplicative Non-commutative proof-nets, in the sequent system of the whole Non-commutative logic, and thus achieve the same kind of efficiency in proof search as in the commutative case. This can be done in three steps:

- First, we have to state the Focusing result in the multiplicative fragment of the Non-commutative sequent system. The only difficulty here is to choose the most appropriate representation for Non-commutative sequents (either with order-varieties or through explicit rules of "See-saw" and "Entropy" - see [12], which shows the equivalence of the two approaches).
- Then, we have to introduce the additive connectives. Their behavior is *a priori* orthogonal to non-commutativity, since removal of the Exchange rule does not affect their commutativity, but we must check that Focusing extends as straightforwardly to the additives as in the commutative case.
- Finally, introducing the exponentials should not cause any major problem: a similar approach to that taken in the commutative case should work, where unbounded formulae are placed in an "extra-territorial area"[2] and can at any time be materialized at any location (in [2], this area is represented in Focusing sequents by an additional field separated by ":").

However, we expect to go beyond this result, and, by analysing thoroughly the invertibility and permutability of inference figures in the Non-commutative case, to achieve a form of proof search optimization which goes beyond the synchronous/asynchronous duality and exploits the specific features of Non-commutativity (Theorem 6 shows that this duality does not distinguish between the commutative and non-commutative cases). In particular, the See-saw and Entropy rules present interesting invertibility properties which are essential to help deciding when to allow them in a Focusing system, preserving the completeness of Focusing while minimizing the intrinsic non-determinism they carry

[2] This expression was originally coined by Jean-Yves Girard, at the Frascati workshop [6]

(reminiscent of the treatment of Weakening and Contraction with Decision rules in the commutative case).

Ultimately, we seek to obtain for Non-commutative Logic a "normal form" analogous to LinLog for Linear Logic, which captures in a restricted, "logic-programming"-like syntax the whole power of Focusing.

Acknowledgement

We would like to thank Michele Abrusci, Paul Ruet and Jean-Yves Girard for fruitful discussions on the topic of this paper, and all the participants at the workshop on Non-commutative Logic, held in Frascati (Italy) in April 1999. This workshop was supported by the TMR network "Linear".

References

1. M. Abrusci and P. Ruet. Non commutative logic I: the multiplicative fragment, 1998. to appear in the *Annals of Pure and Applied Logic.*
2. J-M. Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3), 1992.
3. J-M. Andreoli and R. Pareschi. LO and behold! concurrent structured processes. In *Proc. of OOPSLA/ECOOP'90*, Ottawa, Canada, 1990.
4. J-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
5. J-Y. Girard. On the meaning of logical rules I: syntax vs. semantics II: multiplicatives and additives, 1998. Preprint.
6. J-Y. Girard. Informal communication, 1999. presented at the TMR "Linear" workshop on Non-commutative Logic, Frascati, April 1999.
7. J.S. Hodas and D. Miller. Logic programming in a fragment of intuitionistic linear logic. *Journal of Information and Computation*, 110(2):327–365, 1994.
8. D. Miller. Lexical scoping as universal quantification. In *Proc. of the 6th International Conference on Logic Programming*, Lisboa, Portugal, 1989.
9. D. Miller. Forum: A multiple-conclusion specification logic. *Theoretical Computer Science*, 165(1):201–232, 1996.
10. D. Miller, G. Nadathur, F. Pfenning, and A. Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.
11. C. Retoré. Pomset logic: A non-commutative extension of classical linear logic. In *Proc. of TLCA'97*, pages 300–318, Nancy, France, 1997.
12. P. Ruet. Non commutative logic II: Sequent calculus and phase semantics, 1998. Preprint.

A Demonstrations

A.1 A Focusing Lemma

We first prove the following lemma.

Lemma 1. *Let π be a proof-net with no asynchronous conclusion, and $S = A \otimes B$ be a splitting formula of π. Let π_A, π_B be the two proof nets obtained by splitting π at S. If A is not a negative atom, then*

$$\mathtt{foc}(\pi_A) \setminus \{A\} \quad \subseteq \quad \mathtt{foc}(\pi)$$

(and similarly for the B side)

Demonstration: We proceed by induction on the size of π. Let $F \in \mathtt{foc}(\pi_A) \setminus \{A\}$. Since F is focusing in π_A, there are two cases to consider:

F is a positive atom , and π_A is reduced to an axiom link, with conclusions F and $F^\perp$, one of which being A. But:
- By hypothesis, A is not a negative atom, hence $A \neq F^\perp$.
- By hypothesis, $F \in \mathtt{foc}(\pi_A) \setminus \{A\}$, hence $A \neq F$

Contradiction.

F is a splitting synchronous formula of π_A , of the form $C \otimes D$ and π_A is split at F into two sub-proof-nets π_C, π_D such that
[$\mathcal{P}1$]: C is asynchronous or a negative atom or $C \in \mathtt{foc}(\pi_C)$
[$\mathcal{P}2$]: D is asynchronous or a negative atom or $D \in \mathtt{foc}(\pi_D)$

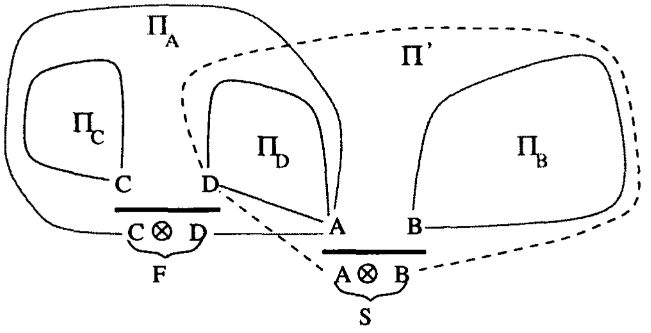


Fig. 4. Different ways of assembling the sub-proof-nets

Since A is a conclusion of π_A different from F and π_A is split at F into π_C, π_D, then A must be in the conclusions of π_C or of π_D. We assume, without loss of generality, that A is a conclusion of π_D (other than D, obviously). Let π' be the proof structure consisting of π_D, π_B and the splitting link of π at S (see Figure 4). It is not difficult to see that

[$\mathcal{P}$.3]: π' is a proof-net split at S into π_D and π_B;

[$\mathcal{P}$.4]: π is split at F into π_C and π'.

Since π' is smaller (in size) than π, we conclude, by the induction hypothesis applied to [$\mathcal{P}$.3], that

$$\texttt{foc}(\pi_D) \setminus \{A\} \subseteq \texttt{foc}(\pi')$$

From this, and [$\mathcal{P}$.2] and $D \neq A$, we infer that

[$\mathcal{P}$.5]: D is asynchronous or a negative atom or $D \in \texttt{foc}(\pi')$

From [$\mathcal{P}$.1] and [$\mathcal{P}$.5] and [$\mathcal{P}$.4], by application of Definition 2, we obtain that $F \in \texttt{foc}(\pi)$.

□

A.2 The Focusing Theorem

We make use of the previous lemma to show Theorem 5:

> Let π be a proof-net containing no asynchronous conclusion. Then $\texttt{foc}(\pi) \neq \emptyset$.

Demonstration: We proceed by contradiction. Let's assume that there exists a proof-net π containing no asynchronous conclusion and such that $\texttt{foc}(\pi) = \emptyset$. We choose π to be of minimal size. We consider two cases:

Either π has no synchronous conclusion , and, since it contains no asynchronous conclusion either, it must be reduced to the axiom link. But then, one of the two conclusions is a positive atom F, which, by Definition 2, is focusing for π. Contradiction.

Or π does contain at least one synchronous conclusion , and, since it contains no asynchronous conclusion, by application of the Splitting lemma, we know that there exists a synchronous conclusion F of π, of the form $A \otimes B$, which splits π into two sub-proof-nets π_A and π_B.

Suppose that

[$\mathcal{P}$.1]: A is neither asynchronous nor a negative atom.

- By construction, the conclusions of π_A other than A are conclusions of π (hence not asynchronous). Since A itself is not asynchronous by [$\mathcal{P}$.1], we infer that none of the conclusions of π_A are asynchronous. Since π_A is strictly smaller than π, which is a proof-net of minimal size without asynchronous nor focusing conclusions, we conclude that

  [$\mathcal{P}$.2]: $\texttt{foc}(\pi_A) \neq \emptyset$
- A is not a negative atom by [$\mathcal{P}$.1], hence, by application of Lemma 1, we have

  [$\mathcal{P}$.3]: $\texttt{foc}(\pi_A) \setminus \{A\} \subseteq \texttt{foc}(\pi)$

Since $\texttt{foc}(\pi) = \emptyset$, we conclude from [$\mathcal{P}$.3] that $\texttt{foc}(\pi_A) \subseteq \{A\}$, and, from [$\mathcal{P}$.2], we conclude that $\texttt{foc}(\pi_A) = \{A\}$. Hence $A \in \texttt{foc}(\pi_A)$.

Thus, by discharging hypothesis [$\mathcal{P}$.1], we conclude

[$\mathcal{P}$.4]: A is asynchronous or a negative atom or $A \in \texttt{foc}(\pi_A)$

By symmetry, we can equally prove that

[$\mathcal{P}$5]: B is asynchronous or a negative atom or $B \in \mathtt{foc}(\pi_B)$

But, from [$\mathcal{P}$4] and [$\mathcal{P}$5], by application of Definition 2, we have that $F \in \mathtt{foc}(\pi)$. Contradiction.

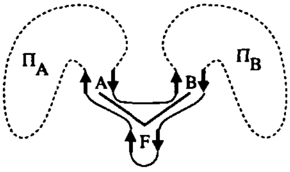
□

A.3 The Splitting Lemma in Non-commutative Logic

We have to show that the Splitting lemma applies to Non-commutative Logic.

> Let π be a non-commutative proof-net with no asynchronous conclusion and at least one synchronous one. Then there exists a synchronous conclusion F such that π consists of two proof nets π_A, π_B plus a synchronous link the premisses of which are conclusions of, resp., π_A and π_B, and the conclusion of which is labeled with F.

<u>**Demonstration**</u>: Let π be a non-commutative proof-net with no asynchronous conclusion and at least one synchronous one. Obviously, π° is a commutative proof-net with no asynchronous conclusion and at least one synchronous one, so is amenable to the commutative splitting lemma (Theorem 3). Hence, π° consists of two proof nets π'_A, π'_B plus a synchronous link the premisses of which are conclusions of, resp., π'_A and π'_B, and the conclusion of which is labeled with F°. By construction of π°, we have that π'_A (resp. π'_B) is of the form π°_A (resp. π°_B), and π consists of π_A, π_B plus a synchronous link the premisses of which are conclusions of, resp., π_A and π_B, and the conclusion of which is labeled with F. Therefore, all we have to check is that π_A and π_B are non-commutative proof-nets (not just proof-structures). In fact, we have that π°_A and π°_B are commutative proof-nets, so, all we have to check is the condition on inner-parts of Theorem 4. Let s_A (resp. s_B) be a ∇3-free switching for π_A (resp. π_B). We can build a ∇3-free switching s for π by assembling s_A, s_B and by choosing the Right switching for F (i.e. $R\otimes$ or $R\odot$ depending on the top-most connective in F):



Let l be a ∇-link of π_A, and let's assume its inner-part in $s_A(\pi_A)$ contains a conclusion C of π_A. There are two cases to consider:

- If C is different from A, then it is a conclusion of π; hence the inner part of l in $s(\pi)$ also contains a conclusion of π. Contradiction (by Theorem 4, since π is a proof-net).
- If $C = A$, then the inner part of l in $s_A(\pi_A)$ goes

$$\cdots, A^{\downarrow}, A^{\uparrow}, \cdots$$

In $s(\pi)$, the inner part of l becomes

$$\cdots, A^{\downarrow}, B^{\uparrow}, \cdots, B^{\downarrow}, F^{\downarrow}, F^{\uparrow}, A^{\uparrow}, \cdots$$

which contains the conclusion F of π. Contradiction (by Theorem 4, since π is a proof-net).

Hence, the inner-part of a ∇-link of π_A in $s_A(\pi_A)$ is the same as that in $s(\pi)$ and does not visit any conclusion of π_A. Since inner-parts of ∇-links do not overlap in $s(\pi)$ (by Theorem 4), neither do they in $s_A(\pi_A)$. □

A.4 The Merging Property

We have to show the following property in Non-commutative Logic (in commutative logic, it is a straightforward consequence of the "no-short-trip" condition over proof-nets).

> Let π_A, π_B be two non-commutative proof-nets. Then the proof structure obtained by assembling π_A, π_B plus a synchronous link the premisses of which are conclusions of, resp., π_A and π_B, is a non-commutative proof net.

Demonstration: Let π_A, π_B be two non-commutative proof-nets and let π be the proof structure obtained by assembling π_A, π_B plus a synchronous link the premisses of which are conclusions A, B of, resp., π_A and π_B. By Theorem 4, we know that π_A° and π_B° are commutative proof-nets, and hence, so is π° (by commutative Merging). Therefore, all we have to prove is that π satisfies the criterion of Theorem 4 on inner-parts. Let s be a ∇3-free switching of π and l be a ∇-link of π. We can assume without loss of generality that l is in π_A. Let s_A be the switching s restricted to π_A. By Theorem 4, we have that the inner-part of l in $s_A(\pi_A)$ contains no conclusion of π_A, and hence does not visit A. Hence the inner-part of l in $s(\pi)$ is the same as that in $s_A(\pi_A)$. Therefore the inner-part of a ∇-link of π in $s(\pi)$ is exactly its inner-part in the sub-proof-net (π_A or π_B) where it occurs. Consequently, since the condition of Theorem 4 holds in these sub-proof-nets, it also holds in π (the non-overlapping condition is obvious if the two links belong to the two different sub-proof-nets). Hence π is a proof-net. □