Technical Communications of the International Conference on Logic Programming, 2010 (Edinburgh), pp. 84–93
http://www.floc-conference.org/ICLP-home.html

FOCUSED PROOF SEARCH FOR LINEAR LOGIC IN THE CALCULUS OF STRUCTURES

NICOLAS GUENOT [1]

[1] Laboratoire d'Informatique (LIX), École Polytechnique
rue de Saclay, 91128 Palaiseau cedex, France
E-mail address: nguenot@lix.polytechnique.fr

Abstract. The proof-theoretic approach to logic programming has benefited from the introduction of *focused* proof systems, through the non-determinism reduction and control they provide when searching for proofs in the sequent calculus. However, this technique was not available in the *calculus of structures*, known for inducing even more non-determinism than other logical formalisms. This work in progress aims at translating the notion of *focusing* into the presentation of *linear logic* in this setting, and use some of its specific features, such as deep application of rules and fine granularity, in order to improve proof search procedures. The starting point for this research line is the multiplicative fragment of linear logic, for which a simple focused proof system can be built.

1. Introduction

The foundational principle of logic programming is the description of computation as a proof search process in some logical deductive system. It is therefore natural to implement logic programming languages in the framework of the sequent calculus, where the theory of proof objects is well-developped and analytic systems are available. This approach has been successful in extending logic programming to richer fragments of logic than Horn clauses, such as hereditary Harrop formulas [Mil91]. However, this required the definition of normal forms for proofs, reducing the search space and providing more structure, first with uniform proofs and then with focused proof systems [And92], initially designed for linear logic and later extended to both intuitionistic and classical logics [Lia09]. The focusing result is now considered an important part of proof theory, and is very much related to the broader and very active study of polarities in linear logic as well as other logics.

Proof Search in the Calculus of Structures. The deep inference methodology has been introduced to overcome the intrisic limitations of the sequent calculus, and design logical formalisms with nice symmetry properties. The most important by-product of this research line is the calculus of structures [Gug07], which generalises the sequent calculus by allowing inference rules to be applied deep inside formulas, as illustrated below:

$$\dfrac{\dfrac{(A \otimes (1 \otimes C)) \parr D}{(A \otimes ((B \parr B^{\perp}) \otimes C)) \parr D}}{(A \otimes (B \parr (B^{\perp} \otimes C))) \parr D}$$

Key words and phrases: proof theory, focusing, proof search, deep inference, linear logic.

Technical Communications of the 26th International Conference on Logic Programming, Edinburgh, July, 2010
Editors: Manuel Hermenegildo, Torsten Schaub
LIPIcs - Leibniz International Proceedings in Informatics. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany
Digital Object Identifier: 10.4230/LIPIcs.ICLP.2010.84

There are several benefits to use this formalism, including the ability to produce proofs exponentially shorter than in the sequent calculus [Bru09], and also a large variety of design choices which allow the definition of several systems for the same logic easily, emphasizing different viewpoints.

However, the freedom given by the calculus of structures in the way of building proofs and using inference rules has an important drawback: the non-determinism involved in the process of proof construction is even greater than it is in the sequent calculus, mostly because of the possible choices regarding the depth of the next inference rule application. Thus, we cannot use directly such a system, and proof search requires the use of specific techniques [Kah06]. The goal of our project is then to get control over non-determinism using the focusing technique, which has to be adapted to this new setting.

Translating the Notion of Focusing. There is no such thing as a formal definition for the notion of focusing. The idea of this technique is mostly induced by the definition of focused sequent calculi, and their proof of completeness, although there is an intuition behind the methodology: invertible inference rules should be applied first, and a sequence of inference rules of the same polarity should always be applied as a group. In order to devise a focused proof system in the calculus of structures, we have to conform to this intuition, so that the result offers *proportionately* the same features as focusing does in the sequent calculus.

Outline of the Paper. We start with a quick survey of the usual focused sequent calculus MLL^F for multiplicative linear logic, and then present the main contribution of this paper, the focused MLS^F system in the calculus of structures, that we prove sound and complete. Finally, we discuss the design of this system, other possible choices as well as extensions to larger fragments of linear logic.

2. The MLL^F Sequent Calculus

We work in the setting of linear logic [Gir87], and more precisely in the multiplicative fragment of this logic. This is a very small and simple logic, where no contraction nor weakening can happen, with two connectives : a conjunction $\otimes$ and a disjunction $\invamp$, dual to each other. Negation can be pushed to the atoms using De Morgan's laws, and we use two sets to represent atoms and their negation, with a bijection $a \mapsto a^\perp$ between them.

The focused sequent calculus MLL^F presented in Figure 1 is a simple restriction of the usual system [And92], where no particular mechanism is required to handle non-linear parts of the context. In this system, simple sequents are replaced with two kinds of sequents, distinguished by the different arrows used to annotate them. Then, inference rules require the definition of three classes of formulas:

$$A_+ ::= A \otimes A \mid 1 \mid a \qquad A_- ::= A \invamp A \mid \bot \mid a^\perp \qquad A_* ::= A_+ \mid a^\perp$$

Remark 2.1. We consider atoms to be positive and their negations to be negative. This is arbitrary, since any *atomic bias* respecting the duality of negation can be used [Mil07].

Annotations are used to enforce that all available $\invamp$ formulas are decomposed before any $\otimes$ formula, and that nested $\otimes$ are treated hereditarily. Another consequence of the syntactic restrictions is that whenever a positive atom is encoutered during a synchronous phase, the identity rule must close the branch. Finally, the management rules of *decision*, *exclusion* and *release* are meant to control annotations, and require to assume that the conclusive sequent of the proof is of the shape $\vdash \Uparrow \Gamma$.

Synchronous Phase **Asynchronous Phase**

$$\otimes\,\frac{\vdash \Gamma \Downarrow A \quad \vdash \Delta \Downarrow B}{\vdash \Gamma, \Delta \Downarrow A \otimes B} \quad \mathsf{id}\,\frac{}{\vdash a^{\perp} \Downarrow a} \quad \mathsf{1}\,\frac{}{\vdash \Downarrow 1} \qquad \parr\,\frac{\vdash \Gamma \Uparrow A, B, \Delta}{\vdash \Gamma \Uparrow A \parr B, \Delta} \quad \bot\,\frac{\vdash \Gamma \Uparrow \Delta}{\vdash \Gamma \Uparrow \bot, \Delta}$$

Management Rules

$$\mathsf{D}\,\frac{\vdash \Gamma \Downarrow A_{+}}{\vdash \Gamma, A_{+} \Uparrow} \quad \mathsf{E}\,\frac{\vdash \Gamma, A_{*} \Uparrow \Delta}{\vdash \Gamma \Uparrow A_{*}, \Delta} \quad \mathsf{R}\,\frac{\vdash \Gamma \Uparrow A_{-}}{\vdash \Gamma \Downarrow A_{-}}$$

Figure 1: Inference rules for system $\mathsf{MLL^F}$

Once this system has been defined, it is necessary to prove that it can be used to replace the usual systems. This is the actual focusing result, which was originally established in the broader case of full linear logic.

Theorem 2.2 (Andreoli, 1992)**.** *The* $\mathsf{MLL^F}$ *system is sound and complete with respect to multiplicative linear logic.*

Searching for proofs in this calculus is much simpler than in unfocused systems, since important points of non-determinism are isolated : context splitting induced by the $\otimes$ rule is a possible point of backtracking, while asynchronous rules cannot induce any backtracking since they are invertible. Moreover, the choice of the next formula to be treated is also controlled using the decision rule, which is the most important source of non-determinism in practice, since context splitting can be implemented in a *lazy* way [Cer00]. The grouping of choices in the synchronous phase reduces the search space, so that the proof construction process is still efficient, while following a uniform strategy.

3. The $\mathsf{MLS^F}$ Focused Calculus of Structures

The starting point for the definition of a focused system in the calculus of structures is the existing system LS for linear logic [Str03], where the multiplicative fragment MLS is very small, using only two inference rules : the *identity* rule $\mathsf{i}{\downarrow}$ and the *switch* rule s. There is no rule for the $\parr$, since there is no *meta-level* connective such as the comma, but we make the treatment of units explicit here, although it is implicit in LS, because we are concerned with proof search. The equational theory only treats associativity and commutativity. Then, a complete proof in LS is a derivation that ends with a 1 alone as a premise.

Adapting the focusing technique to the calculus of structures requires to define a syntax that enforces the correct shape for proofs through annotations added to the usual system. The resulting system is called $\mathsf{MLS^F}$, and its equational theory and inference rules are given in Figure 2. Beyond annotations, it uses the same classes of formulas as $\mathsf{MLL^F}$. It also uses a global flag, that can be either 0 or 1 to indicate that we are in a *decision* or *focusing* phase respectively. Indeed, the structure of phases is different from the sequent calculus: the asynchronous and synchronous phases are merged into the focusing phase, and the decision phase corresponds to the application of the decision rule.

Syntactic Congruence

$$\begin{array}{rcl} A \parr B & \equiv & B \parr A \\ A \otimes B & \equiv & B \otimes A \\ & & \\ A \parr (B \parr C) & \equiv & (A \parr B) \parr C \\ A \otimes (B \otimes C) & \equiv & (A \otimes B) \otimes C \end{array}$$

the flag $n = 0/1$ on a structure indicates the phase it belongs to, by reflecting the number of down arrows in the structure.

Focusing Phase

$$\mathsf{s}\,\frac{\xi\{[A_* \parr \Downarrow B] \otimes C)\}_1}{\xi\{A_* \parr \Downarrow (B \otimes C)\}_1} \qquad \mathsf{ai}\,\frac{\xi\{\Downarrow 1\}_1}{\xi\{a^\perp \parr \Downarrow a\}_1}$$

$$\mathsf{f}\,\frac{\xi\{\Downarrow A\}_1}{\xi\{\perp \parr \Downarrow A\}_1} \qquad \mathsf{b}\,\frac{\xi\{A\}_0}{\xi\{\Downarrow 1 \otimes A\}_1}$$

Management Rules

$$\mathsf{d}\,\frac{\xi\{\Downarrow A_+\}_1}{\xi\{A_+\}_0} \qquad \mathsf{r}\,\frac{\xi\{A_-\}_0}{\xi\{\Downarrow A_-\}_1}$$

Figure 2: Congruence and inference rules for system $\mathsf{MLS^F}$

Although it is built on the same idea as focused proof systems in the sequent calculus, there are several important differences, due to the deep inference setting, that should be explained concerning $\mathsf{MLS^F}$ and its proof construction dynamics:

- The proof search process is a sequence of focusing phases, delimited by applications of the decision rule d. A focusing phase represents the interaction between a positive formula and another formula in its surrounding context. As would be done during an asynchronous phase, some $\perp$ formulas can be erased in such a focusing phase.
- The *switch* rule s implements lazy splitting within the proof-theoretic framework, so that branching is decomposed into the stepwise distribution of formulas in the context to different sides of a $\otimes$ connective. Thus, formulas are moved hereditarily within a compound $\otimes$ structure, and we obtain a *uniform translation* of the focusing idea into the calculus of structures, in the sense that it provides proportionately the same restriction as in the sequent calculus, although it is a weaker normal form.
- An important consequence of the focusing result is the natural definition of *synthetic connectives* [Zei08]. Here, the definition of positive synthetic connectives is clear, but there is no equivalent on the negative side. However, this corresponds to the asymmetry of focusing, that emphasizes the importance of the positives. Indeed, the reason we decompose negatives first is simply that a part of them should be treated before the next positive connective to be chosen. It is even impossible to observe negative synthetic connectives in the conclusive sequent of a proof, since they are built during the proof construction process. The $\mathsf{MLS^F}$ system acknowledges this fact by orienting proof search with the help of positives only, and defining deduction steps in terms of interaction between a positive formula and its context.

Soundness is a trivial result for such a system, since all of its rules are sound once the annotations have been removed. Then, completeness can be proved by using a translation from proofs in $\mathsf{MLL^F}$ to proofs in our system in the calculus of structures. This requires to define a translation from both kinds of sequents to structures with annotations.

Definition 3.1. The translation $\llbracket \cdot \rrbracket_{\mathsf{S}}$ from $\mathsf{MLL}^{\mathsf{F}}$ sequents to $\mathsf{MLS}^{\mathsf{F}}$ structures is defined using the translation $\llbracket \cdot \rrbracket_{\mathsf{F}}$ from multisets of formulas to structures, as follows:

$$\begin{array}{l} \llbracket \vdash \Gamma \Uparrow \Delta \rrbracket_{\mathsf{S}} = \llbracket \Gamma \rrbracket_{\mathsf{F}} \parr \llbracket \Delta \rrbracket_{\mathsf{F}} \\ \llbracket \vdash \Gamma \Downarrow A \rrbracket_{\mathsf{S}} = \llbracket \Gamma \rrbracket_{\mathsf{F}} \parr \Downarrow \llbracket A \rrbracket_{\mathsf{F}} \end{array} \qquad \llbracket A_1, \cdots, A_n \rrbracket_{\mathsf{F}} = A_1 \parr \cdots \parr A_n$$

The translation of sequent calculus focused proofs into our system is not as easy as the translation of formulas. Indeed, the differences in the way phases are organised and the decomposition of $\otimes$ splitting induce a different shape of proofs. Therefore, we have to use an intermediate system, called $\mathsf{MLS}^{\mathsf{F}}_{\mathsf{s}}$, where the s and f rules are replaced with the following variants, closer to the sequent calculus:

$$\mathsf{f_s}\,\frac{\xi\{A\}_0}{\xi\{\bot \parr A\}_0} \qquad \mathsf{s_s}\,\frac{\xi\{[A_\bullet \parr \Downarrow B] \otimes C)\}_1}{\xi\{A_\bullet \parr \Downarrow (B \otimes C)\}_1} \qquad \textit{where} \quad A_\bullet ::= A_* \mid A_* \parr A_\bullet$$

This system relaxes the restriction on the f rule, so that $\bot$ can be erased anywhere, between two focusing phases, and the switch rule is more general. We now prove completeness for this system and we will then show how to translate these proofs into $\mathsf{MLS}^{\mathsf{F}}$ proofs.

Lemma 3.2. *For any proof a sequent* $\vdash \mathcal{M}$ *in* $\mathsf{MLL}^{\mathsf{F}}$*, there is a proof of* $\llbracket \vdash \mathcal{M} \rrbracket_{\mathsf{S}}$ *in* $\mathsf{MLS}^{\mathsf{F}}_{\mathsf{s}}$*.*

Proof. By translation of a given proof Π in $\mathsf{MLL}^{\mathsf{F}}$ to a proof in the focused system $\mathsf{MLS}^{\mathsf{F}}_{\mathsf{s}}$, using a case analysis and an induction on the height of this proof tree. The base case happens when translating axiomatic rule instances, for which the resulting derivation should end with the premise $\Downarrow 1$.

(i) The case of a $\mathsf{1}$ rule is immediate, since the conclusion is $\Downarrow \mathsf{1}$.

(ii) An identity rule id is directly translated as an identity ai rule:

$$\mathsf{id}\,\frac{}{\vdash a^\perp \Downarrow a} \quad \longrightarrow \quad \mathsf{ai}\,\frac{(\Downarrow \mathsf{1})_1}{[a^\perp \parr \Downarrow a]_1}$$

The inductive cases use the proofs produced by applying the induction hypothesis, which can be plugged into the derivation being build, because of the deep inference setting — and this is done in accordance to the global flag. In the following, we make explicit the use of the syntactic congruence by using a fake $\equiv$ rule whenever a structure is rewritten into another. Moreover, we must be cautious when translating the $\otimes$ rule since there are two different cases, depending on the polarity of the formulas on both sides of the $\otimes$ connective being decomposed.

(iii) If there is at least one positive structure on one side of the $\otimes$, we can use it to apply a decision d rule in the end of the proof:

$$\otimes\,\frac{\overset{\Pi_A}{\vdash \Gamma \Downarrow A} \qquad \overset{\Pi_B}{\vdash \Delta \Downarrow B_+}}{\vdash \Gamma, \Delta \Downarrow A \otimes B_+} \quad \longrightarrow \quad \begin{array}{c} \mathsf{b}\,\cfrac{\mathsf{d}\,\cfrac{\overset{\Lambda_B \Vert}{[\Delta \parr \Downarrow B_+]_1}}{[\Delta \parr B_+]_0}}{[\Delta \parr (\Downarrow \mathsf{1} \otimes B_+)]_1} \\ \Lambda_A \Big\Vert \\ \mathsf{s_s}\,\cfrac{[\Delta \parr ([\Gamma \parr \Downarrow A] \otimes B_+)]_1}{[\Gamma \parr \Delta \parr \Downarrow (A \otimes B_+)]_1} \end{array}$$

(iv) In the other case, there are negative structures on both sides of the $\otimes$, and focusing enforces the use of the release rule:

$$\otimes\,\dfrac{\mathsf{R}\,\dfrac{\begin{array}{c}\Pi_A\\ \vdash\Gamma\Uparrow A_-\end{array}}{\vdash\Gamma\Downarrow A_-}\qquad \mathsf{R}\,\dfrac{\begin{array}{c}\Pi_B\\ \vdash\Delta\Uparrow B_-\end{array}}{\vdash\Delta\Downarrow B_-}}{\vdash\Gamma,\Delta\Downarrow A_-\otimes B_-}\quad\longrightarrow\quad \begin{array}{c}\Lambda_B\Vert\\ \mathsf{b}\,\dfrac{[\Delta\parr B_-]_0}{[\Delta\parr(\Downarrow 1\otimes B_-)]_1}\\ \Lambda_A\Vert\\ \mathsf{s_s}\,\dfrac{\mathsf{r}\,\dfrac{[\Delta\parr([\Gamma\parr A_-]\otimes B_-)]_0}{[\Delta\parr([\Gamma\parr\Downarrow A_-]\otimes B_-)]_1}}{[\Gamma\parr\Delta\parr\Downarrow(A_-\otimes B_-)]_1}\end{array}$$

(v) The $\parr$ rule is not really translated since it has no effect on the translation:

$$\parr\,\dfrac{\begin{array}{c}\Pi'\\ \vdash\Gamma\Uparrow A,B,\Delta\end{array}}{\vdash\Gamma\Uparrow A\parr B,\Delta}\quad\longrightarrow\quad \begin{array}{c}\Lambda'\Vert\\ \equiv\,\dfrac{[\Gamma\parr A\parr B\parr\Delta]_0}{[\Gamma\parr[A\parr B]\parr\Delta]_0}\end{array}$$

(vi) The $\bot$ rule is simply translated as a $\mathsf{f_s}$ rule:

$$\parr\,\dfrac{\begin{array}{c}\Pi'\\ \vdash\Gamma\Uparrow\Delta\end{array}}{\vdash\Gamma\Uparrow\bot,\Delta}\quad\longrightarrow\quad \begin{array}{c}\Lambda'\Vert\\ \mathsf{f_s}\,\dfrac{[\Gamma\parr\Delta]_0}{[\Gamma\parr\bot\parr\Delta]_0}\end{array}$$

(vii) The decision rules D and d are used the same way in both systems:

$$\mathsf{D}\,\dfrac{\begin{array}{c}\Pi'\\ \vdash\Gamma\Downarrow A_+\end{array}}{\vdash\Gamma,A_+\Uparrow}\quad\longrightarrow\quad \begin{array}{c}\Lambda'\Vert\\ \mathsf{d}\,\dfrac{[\Gamma\parr\Downarrow A_+]_1}{[\Gamma\parr A_+]_0}\end{array}$$

(viii) The exclusion rule E is not translated at all, since not required:

$$\mathsf{E}\,\dfrac{\begin{array}{c}\Pi'\\ \vdash\Gamma,A_*\Uparrow\Delta\end{array}}{\vdash\Gamma\Uparrow A_*,\Delta}\quad\longrightarrow\quad \begin{array}{c}\Lambda'\Vert\\ [\Gamma\parr A_*\parr\Delta]_0\end{array}$$

(ix) The release rules R and r are also used the same way in both systems:

$$\mathsf{R}\,\dfrac{\begin{array}{c}\Pi'\\ \vdash\Gamma\Uparrow A_-\end{array}}{\vdash\Gamma\Downarrow A_-}\quad\longrightarrow\quad \begin{array}{c}\Lambda'\Vert\\ \mathsf{r}\,\dfrac{[\Gamma\parr A_-]_0}{[\Gamma\parr\Downarrow A_-]_1}\end{array}$$

■

Finally, we can establish the focusing result, which states that the $\mathsf{MLS^F}$ proof system is sound and complete, by using our translations. Unfortunately, this result does not allow a comparison with the sequent calculus on the level of focusing, since it is difficult to give a formal account of the reduction of the proof search space induced by both systems. The benefits of our focusing restriction are yet to be further studied.

Theorem 3.3. $\mathsf{MLS^F}$ *is sound and complete with respect to multiplicative linear logic.*

Proof. Soundness is immediately obtained by removing arrows from inference rules in $\mathsf{MLS^F}$, leaving us with the rules of MLS, and trivialised management rules with the same structure as premise and conclusion. Completeness is obtained in two steps. First, given a sequent calculus proof Π of $\vdash \mathcal{M}$ we produce a proof Λ of $[\![\vdash \mathcal{M}]\!]_{\mathsf{S}}$ in $\mathsf{MLS^F_s}$ using Lemma 3.2. Then we have to produce a proof of $[\![\vdash \mathcal{M}]\!]_{\mathsf{S}}$ in $\mathsf{MLS^F}$, by modifying Λ.

Because of the shape of sequent calculus proofs, all instances of $\mathsf{f_s}$ in Λ are located below an instance of d in the surrounding context. It is thus possible to permute all instances of $\mathsf{f_s}$ above the associated decision rule, turning them into instances of f, as follows:

$$\mathsf{f_s}\,\dfrac{\mathsf{d}\,\dfrac{\xi\{\Gamma \parr \Downarrow A_+ \parr \Delta\}_1}{\xi\{\Gamma \parr A_+ \parr \Delta\}_0}}{\xi\{\Gamma \parr \bot \parr A_+ \parr \Delta\}_0} \quad\longrightarrow\quad \mathsf{d}\,\dfrac{\mathsf{f}\,\dfrac{\xi\{\Gamma \parr \Downarrow A_+ \parr \Delta\}_1}{\xi\{\Gamma \parr \bot \parr \Downarrow A_+ \parr \Delta\}_1}}{\xi\{\Gamma \parr \bot \parr A_+ \parr \Delta\}_0}$$

Then, we have to reorganise sequences of $\mathsf{s_s}$ instances into sequences of s instances. Again, because of the shape of sequent calculus proofs, we know that instances of $\mathsf{s_s}$, if they cannot be read as simple instances of s, follow a precise scheme where the reaction rule is applied above a sequence of $\mathsf{s_s}$ instances. We can thus rewrite the derivation as follows:

$$\mathsf{s_s^*}\,\dfrac{\mathsf{r}\,\dfrac{\xi\{\zeta\{A \parr B_* \parr C_-\}\}_1}{\xi\{\zeta\{A \parr B_* \parr \Downarrow C_-\}\}_1}}{\xi\{[A \parr B_*] \parr \Downarrow\zeta\{C_-\}\}_1} \quad\longrightarrow\quad \mathsf{s^*}\,\dfrac{\mathsf{r}\,\dfrac{\mathsf{d}\,\dfrac{\mathsf{s_s^*}\,\dfrac{\mathsf{r}\,\dfrac{\xi\{\zeta\{A \parr B_* \parr C_-\}\}_0}{\xi\{\zeta\{A \parr \Downarrow[B_* \parr C_-]\}\}_1}}{\xi\{A \parr \Downarrow\zeta\{B_* \parr C_-\}\}_1}}{\xi\{A \parr \zeta\{B_* \parr C_-\}\}_0}}{\xi\{A \parr \zeta\{B_* \parr \Downarrow C_-\}\}_1}}{\xi\{[A \parr B_*] \parr \Downarrow\zeta\{C_-\}\}_1}$$

By a simple induction on the size of the structure being moved deep inside the context $\zeta\{\cdot\}$, we can replace all instances of $\mathsf{s_s}$ with instances of s. Therefore, Λ can be turned into a proof Λ' of $[\![\vdash \mathcal{M}]\!]_{\mathsf{S}}$ in $\mathsf{MLS^F}$. ■

4. Variations and Extensions

The system we devised corresponds to one possible design choice among many others. It has been chosen because of its simplicity, and its use of the important feature of lazy splitting introduced by the deep inference methodology. This is probably an important feature from the viewpoint of logic programming, since it allows for clean proof-theoretic foundations for the context sharing implementation technique. However, it is unclear which are the features that could further improve proof search procedures with respect to this programming consideration, and if they would be compatible with the design choices we made so far. We discuss now some variations of the $\mathsf{MLS^F}$ system, as well as its extension to larger fragments of the logic and stronger normal forms.

Complete ⊗ Decomposition. An alternative to the current design of the switch rule s is the use of a maximality condition, imposing that all elements of the immediate surrounding context are moved inside the ⊗ structure, along with a one-step splitting variation of the switch rule, as follows:

$$\mathsf{s}\,\frac{\xi\{[A \parr \Downarrow C] \otimes [B \parr \Downarrow D]\}_2}{\xi\{[A \parr B] \parr \Downarrow(C \otimes D)\}_1}$$

However, this would require to extend the global flag to a general counter, which would track the number of down arrows in the structure. More importantly, this would betray the idea of small-step rules decomposition that comes as a benefit when using the calculus of structures. Such a complete decomposition does not fit the deep inference methodology, since it is impossible to obtain negative synthetic connectives using this approach, unless we mimick sequent calculus proofs.

Separate Asynchronous Phase. It is possible to use another annotation to control the asynchronous rules, thus splitting the focusing phase into asynchronous and synchronous phases, as it is done in the sequent calculus. However, the asynchronous phase would still be directed by the future synchronous phase, since we must ensure that all required negative structures have been treated before we start treating a positive. Such a design would require to relate these two new phases through the interaction of annotated formulas, as can be done with the following alternative rules:

$$\mathsf{s}\,\frac{\xi\{[\Uparrow A_* \parr \Downarrow B] \otimes C)\}}{\xi\{\Uparrow A_* \parr \Downarrow(B \otimes C)\}} \qquad \mathsf{ai}\,\frac{\xi\{\Downarrow 1\}}{\xi\{\Uparrow a^\perp \parr \Downarrow a\}} \qquad \mathsf{f}\,\frac{\xi\{A\}}{\xi\{\Uparrow \perp \parr A\}}$$

With such an extension of the annotations, we have to extend the global flag, so that the presence of up arrows is tracked too. This complicates the managment rules, and does not provide any benefit compared to MLS^F if we only allow for one up arrow in a structure. Then, multiple up arrows would allow for a compromise between small-step and big-step splitting of the ⊗, but it requires to use extend the use of down arrows too. In order to allow medium-step splitting, it seems simpler to use the $\mathsf{MLS}^\mathsf{F}_\mathsf{s}$ intermediate system used in the completeness proof, where a group of structures can be moved inside a positive.

Extension to the Additives. Using the additive connectives & and ⊕ in the setting of our focused calculus of structures yields new design choices that should be studied carefully. An intuitive way of extending the system would be to add the two following rules:

$$\mathsf{a}\,\frac{\xi\{[\Downarrow A \parr B] \with [\Downarrow A \parr C]\}_2}{\xi\{\Downarrow A \parr (B \with C)\}_1} \qquad \mathsf{o}\,\frac{\xi\{\Downarrow A\}_1}{\xi\{\Downarrow [A \oplus B]\}_1}$$

The focused rule for the ⊕ is perfectly fine, but the rule for & is more problematic. Indeed, the duplication of the down arrow implies that the global flag has to count these arrows, as considered for the alternate switch rule. Moreover, the treatment of the left and right branches of the & are now interleaved, although a unique active positive structure provides us with a stronger normal form. Therefore, the following rule should be considered:

$$\mathsf{a}\,\frac{\xi\{[\Downarrow A \parr B] \with [A \parr C]\}_1}{\xi\{\Downarrow A \parr (B \with C)\}_1}$$

This rule corresponds to the viewpoint of *slices* on the additives, each slice being treated separately. Moreover, this fits the idea that a focusing phase represents the interaction between a positive structure and only one of the element of its context — its context being more structured here than a simple multiset created by ⅋.

Multi-focusing. The usual notion of focusing can be extended to provide a stronger normal form, by handling multiple focused formulas [Cha08]. This idea seems quite natural in the setting of $\mathsf{MLS}^{\mathsf{F}}$, where the global flag can be extended into a counter, as mentioned above. There are three different ways of using multiple arrows in such a system:

- *Parallel focusing:* when arrows are located in different subformulas of a conjunction connective, they represent the parallel focusing of different branches in the sequent calculus. It is easy to handle multiple arrows in this case, but it is unclear how this could yield a stronger normal form.
- *Multi-focusing:* the use of multiple focusing arrows with the same $\parr$ context is more tricky. This is the case corresponding to the notion of multi-focusing in the sequent calculus, and it is difficult to know if two given positive structures can be focused at the same time — naïve maximal multi-focusing is not complete.
- *Nested focusing:* it seems easy to handle cases where an arrow annotates a positive structure located deep inside another focused structure. Moreover, this suggests an extension of synthetic connectives to disconnected layers of positive connectives, which could be interesting in the search for a stronger normal form.

5. Conclusion and Future Work

The $\mathsf{MLS}^{\mathsf{F}}$ system presented here is a first step in the larger project of extracting the focusing notion out of its sequent calculus roots. This is a prototype that should be further studied and improved, since although we know there are some benefits in using the calculus of structures as an host formalism for proof search and logic programming — for instance, lazy splitting and shorter proofs —, it is yet unclear how to gain control over its overwhelming non-determinism, especially for larger logics, such as full linear logic or even classical logic [Brü03].

The next step in this research project is to design a focused proof system for the multiplicative-additive fragment of linear logic in the calculus of structures, and solve there the problem of controlling duplications of focused formulas. Then, it might not be more difficult to accomodate the exponentials, since they do not quite fit the focusing categories in the sequent calculus — they are located at the interface between polarity groups.

In order to assess the benefits of the deep inference methodology in the definition of proof search procedures, it also seems necessary to experiment with our system. Thus, a prototype implementation of such a focused system should be carried out, so that it can be compared to other proof search software developed on top of the focusing approach [Bae07]. Choosing among the many possible designs requires to get experience with proof search in the calculus of structures, and it might even be needed to develop specific variants of the usual focusing, so that the dynamics of deep inference proof search can be used with full power — indeed, it would be useless to try to mimick the proof search behaviour of the sequent calculus, while the benefits are in the proofs that cannot be written in a shallow way. Finally, the study of deep inference proof methods raises the question of its status: is it only a way to implement more efficient proof search procedures with clean semantics, or will it yield new features to be used in logic programming languages ?

Acknowledgements

This work has benefited from useful comments and suggestions of the anonymous reviewers, and fruitful discussions with Kaustuv Chaudhuri about the design of focused systems.

References

[And92] J-M. Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

[Bae07] D. Baelde and D. Miller. Least and greatest fixed points in linear logic. In N. Dershowitz and A. Voronkov (eds.), *LPAR'07*, *LNCS*, vol. 4790, pp. 92–106. 2007.

[Brü03] K. Brünnler. *Deep Inference and Symmetry in Classical Proofs*. Ph.D. thesis, Technische Universität Dresden, 2003.

[Bru09] P. Bruscoli and A. Guglielmi. On the proof complexity of deep inference. *ACM Transactions on Computational Logic*, 10(2):1–34, 2009.

[Cer00] I. Cervesato, J. S. Hodas, and F. Pfenning. Efficient resource management for linear logic proof search. *Theoretical Computer Science*, 232(1-2):133–163, 2000.

[Cha08] K. Chaudhuri, D. Miller, and A. Saurin. Canonical sequent proofs via multi-focusing. In G. Ausiello, J. Karhumäki, G. Mauri, and L. Ong (eds.), *Fifth IFIP International Conference on Theoretical Computer Science*, vol. 273, pp. 383–396. 2008.

[Gir87] J-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Gug07] A. Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64, 2007.

[Kah06] O. Kahramanoğulları. *Nondeterminism and Language Design in Deep Inference*. Ph.D. thesis, Technische Universität Dresden, 2006.

[Lia09] C. Liang and D. Miller. Focusing and polarization in linear, intuitionistic, and classical logics. *Theoretical Computer Science*, 410(46):4747–4768, 2009.

[Mil91] 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.

[Mil07] D. Miller and A. Saurin. From proofs to focused proofs : a modular proof of focalization in linear logic. In J. Duparc and T. A. Henzinger (eds.), *CSL'07*, *LNCS*, vol. 4646, pp. 405–419. 2007.

[Str03] L. Straßburger. *Linear Logic and Noncommutativity in the Calculus of Structures*. Ph.D. thesis, Technische Universität Dresden, 2003.

[Zei08] N. Zeilberger. Focusing and higher-order abstract syntax. In G. Necula and P. Wadler (eds.), *POPL'08*, pp. 359–369. 2008.