

The Focused Calculus of Structures

Kaustuv Chaudhuri, Nicolas Guenot, and Lutz Straßburger

INRIA & LIX/École Polytechnique
Route de Saclay, 91128 Palaiseau, France
{kaustuv,nguenot,lutz}@lix.polytechnique.fr

Abstract

The *focusing* theorem identifies a complete class of sequent proofs that have no inessential non-deterministic choices and restrict the essential choices to a particular normal form. Focused proofs are therefore well suited both for the search and for the representation of sequent proofs. The *calculus of structures* is a proof formalism that allows rules to be applied deep inside a formula. Through this freedom it can be used to give analytic proof systems for a wider variety of logics than the sequent calculus, but standard presentations of this calculus are too permissive, allowing too many proofs. In order to make it more amenable to proof search, we transplant the focusing theorem from the sequent calculus to the calculus of structures. The key technical contribution is an incremental treatment of focusing that avoids trivializing the calculus of structures. We give a direct inductive proof of the completeness of the focused calculus of structures with respect to a more standard unfocused form. We also show that any focused sequent proof can be represented in the focused calculus of structures, and, conversely, any proof in the focused calculus of structures corresponds to a focused sequent proof.

1998 ACM Subject Classification F.4.1 Mathematical Logic: Proof theory

Keywords and phrases Focusing, Polarity, Calculus of Structures, Linear Logic

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Logic has traditionally been seen as a means of representing and systematizing mathematical knowledge, but it is increasingly being used to encode and reason about formal systems—programming languages, process calculi, transition systems, *etc.*—that are inherently computational. In this use of logic, the syntax of proofs is important to build correspondences between the proofs in logic and the computations of the encoded systems, also known as the problem of *representational adequacy*. An adequate encoding is not only manifestly correct, *i.e.*, it represents all and only the computations of the encoded system, but is also useful as a device to automate the reasoning in and about the encoded system. In standard proof systems such as Gentzen’s *sequent calculus*, it is usually impossible to construct adequate encodings: there are more proofs than computational traces, because the inference rules are more non-deterministic than the computational steps.

In recent years the *focusing* theorem of Andreoli [1] has been used to create certain “normal forms” of sequent proofs where the question of representational adequacy becomes considerably easier, often trivial, for focused proofs. Focusing was originally developed for (classical) linear logic but has since been extended to a wide spectrum of logics [3, 14]. The essential observation of focusing is that sequent rules have certain natural permutative affinities that can be exploited to fuse logical connectives into larger *synthetic connectives*; for example, the synthetic connective $- \otimes (- \oplus -)$ behaves as a ternary connective instead of as a composition of two binary connectives. The problem of representational adequacy



© Kaustuv Chaudhuri, Nicolas Guenot, and Lutz Straßburger;
licensed under Creative Commons License ND

20th Annual EACSL Conference on Computer Science Logic.

Editor: Marcus Bezem; pp. 1–15



Leibniz International Proceedings in Informatics
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

is reduced to that of encoding the computational steps in such a way that they correspond exactly to the synthetic connectives.

This technique has been successful for the standard classical, intuitionistic, and linear logics where the sequent calculus is most natural. The sequent calculus is, however, inherently limited in its expressivity: it cannot be used to give *analytic* (*i.e.*, cut-free) proof systems for many modal and non-commutative logics that have been used for program safety, operational semantics, or linguistics. The common feature of many of these logics is that they rely on *deep inference* or the ability to perform deduction inside a formula. Proof systems for many such logics need to generalize the sequent calculus; some popular generalizations include: hypersequents [2], nested sequents [5], or the display calculus [4]. The most permissive, and therefore most expressive, of such generalizations is the *calculus of structures* [10, 11, 6] that does not differentiate between formulas and sequents and can therefore perform deduction anywhere inside a formula. Besides the increased expressivity, proofs in the calculus of structures can also exploit features that are not available in the other shallower formalisms; for example, they can be exponentially smaller than sequent proofs [7], or they can be decomposed in a number of ways [20].

A main distinguishing feature of the calculus of structures is that it divides the sequent rules into smaller components, thereby introducing more non-deterministic choices. The sequent calculus operates on entire (multi-)sets of formulas, with a single sequent rule able to split whole contexts multiplicatively or test for the absence of certain elements. The rules of the calculus of structures, on the other hand, perform such operations incrementally on fragments of contexts. Some of the inessential choices introduced by this incremental nature can be removed by restricting the *syntactic congruence* in the original formulation of the calculus [10, 11], leading to the system LS for classical propositional linear logic (outlined in Section 3.1), which can be seen as a variant of the original formulation in [17, 18] that is more amenable to automation. One important feature of LS is that contraction is the sole rule that makes proofs unbounded, and it permutes below all other rules (Proposition 8), a property that is crucial for the cut-elimination result for LS (also presented in Section 3.1). The contraction-free fragment of LS is therefore decidable.

Yet, despite its more parsimonious design, LS is still at least as non-deterministic as the unfocused sequent calculus. It is natural to ask if a result similar to focusing can tame LS in the same way that the sequent system LLK (without cut) for linear logic (Figure 1) was tamed to produce the focused system LLKF (Figure 2). For the purely multiplicative fragment, this question has already been investigated in [9], but the strategy there seems difficult to generalize. In this paper (in Section 3.2) we construct a focused variant of the calculus LS, called LSF, for full classical propositional linear logic. It uses the technical device of polarized formulas [12]; polarities make the synthetic connectives manifest in the syntax, and the rules of LSF are organized to respect polarity, *i.e.*, to never introduce a polarity change that did not already exist. Synthetic connectives are thus preserved in LSF proofs.

To show LSF complete with respect to LS in its own right, *i.e.*, that any LS proof can be turned into an LSF proof, we build a equivalent synthetic variant of LSF called LSS (see Section 3.2). A special rule that breaks the polarity restriction is added to LSS to represent unfocused LS proofs directly, and then this rule is shown to be admissible in LSS. We thus have a simple *internal* proof of completeness of LSS (and hence of LSF) with respect to LS. This style of showing completeness of focusing for the calculus of structures can pave the way for focused variants of other logics that lack an analytic sequent system. Although we limit our attention to classical propositional linear logic in this paper, we consider it an important future work to extend our focusing result to logics for which focusing in terms of

$$\begin{array}{c}
\text{id} \frac{}{\vdash a, \bar{a}} \quad \text{ct} \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A} \quad \text{wk} \frac{\vdash \Gamma}{\vdash \Gamma, ?A} \quad \text{cut} \frac{\vdash \Gamma, A \quad \vdash \Delta, (A)^\perp}{\vdash \Gamma, \Delta} \\
\otimes \frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} \quad 1 \frac{}{\vdash 1} \quad \oplus_1 \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B} \quad \oplus_2 \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B} \quad ! \frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A} \\
\wp \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \wp B} \quad \perp \frac{\vdash \Gamma}{\vdash \Gamma, \perp} \quad \& \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B} \quad \top \frac{}{\vdash \Gamma, \top} \quad ? \frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}
\end{array}$$

■ **Figure 1** LLK: a one-sided single-zoned sequent calculus for classical propositional linear logic

the sequent calculus is inapplicable. This includes logics like BV [10], the logic of bunched implications [16], and various modal logics.

We also compare LSF and LLKF by first showing that any LLKF proof can be simulated in LSF (in Section 4.1), *i.e.*, that LSF is powerful enough to represent focused sequent proofs. Then we also give an algorithm to *extract* an LLKF proof from any LSF proof that is unique up to permutations between negative rules (in Section 4.2). These two results justify the use of the adjective “focused” for LSF. Together with the completeness of LSF for LS, this result can be used to give an alternative proof of completeness of LLKF for LLK.

2 The Sequent Calculus and Focusing for Linear Logic

We begin with a quick overview of the standard sequent calculus and the focusing theorem for classical propositional linear logic whose formulas (A, B, \dots) have the following grammar:

$$A, B ::= a \mid A \otimes B \mid 1 \mid A \oplus B \mid 0 \mid !A \mid \bar{a} \mid A \wp B \mid \perp \mid A \& B \mid \top \mid ?A$$

The *atoms* (a, b, \dots) are drawn from some countably infinite set. Formulas are in negation normal form, with the negation of a written as \bar{a} , and negation of formulas $(-)^{\perp}$ as follows:

$$\begin{aligned}
(a)^{\perp} &= \bar{a} & (A \otimes B)^{\perp} &= (A)^{\perp} \wp (B)^{\perp} & (1)^{\perp} &= \perp & (A \oplus B)^{\perp} &= (A)^{\perp} \& (B)^{\perp} & (0)^{\perp} &= \top & (!A)^{\perp} &= ?(A)^{\perp} \\
(\bar{a})^{\perp} &= a & (A \wp B)^{\perp} &= (A)^{\perp} \otimes (B)^{\perp} & (\perp)^{\perp} &= 1 & (A \& B)^{\perp} &= (A)^{\perp} \oplus (B)^{\perp} & (\top)^{\perp} &= 0 & (?A)^{\perp} &= !(A)^{\perp}
\end{aligned}$$

The standard sequent calculus for linear logic, called LLK and shown in Figure 1, is given in terms of one-sided single-zoned *sequents* of the form $\vdash \Gamma$ where Γ is a *context* (a multi-set of formulas).

► **Theorem 1** (cut elimination). *The cut rule is admissible in $\text{LLK} \setminus \{\text{cut}\}$.* ◀

Let us now sketch the focused variant of LLK, called LLKF. For this, the formulas are divided into two *polarity* classes—*positive* and *negative*—based on the permutation properties of their sequent rules. The negative formulas have invertible rules, *i.e.*, rules that may be applied whenever the formula occurs in the context, while the rules for the positive formulas are sensitive to the order of application of rules and are therefore generally non-invertible. Following [12], we syntactically distinguish these two classes and mediate between them by a pair of *shift* connectives (\downarrow and \uparrow):

$$\begin{aligned}
P, Q &::= a \mid P \otimes Q \mid 1 \mid P \oplus Q \mid 0 \mid !N \mid \downarrow N && \text{(Positive Formulas)} \\
N, M &::= \bar{a} \mid N \wp M \mid \perp \mid N \& M \mid \top \mid ?P \mid \uparrow P && \text{(Negative Formulas)}
\end{aligned}$$

Here, $(\downarrow N)^{\perp} = \uparrow(N)^{\perp}$ and $(\uparrow P)^{\perp} = \downarrow(P)^{\perp}$. We use the following contexts for LLKF:

Structural				
$\text{wdc} \frac{\vdash \Gamma, P; \Pi, [P]}{\vdash \Gamma, P; \Pi}$	$\text{dc} \frac{\vdash \Gamma; \Pi, [P]}{\vdash \Gamma; \Pi, \uparrow P}$	$\downarrow \frac{\vdash \Gamma; \Pi, N}{\vdash \Gamma; \Pi, [\downarrow N]}$		
Positive phase				
$\text{id} \frac{}{\vdash \Gamma; \bar{a}, [a]}$	$\otimes \frac{\vdash \Gamma; \Pi_1, [P] \quad \vdash \Gamma; \Pi_2, [Q]}{\vdash \Gamma; \Pi_1, \Pi_2, [P \otimes Q]}$	$1 \frac{}{\vdash \Gamma; [1]}$		
$\oplus \frac{\vdash \Gamma; \Pi, [P]}{\vdash \Gamma; \Pi, [P \oplus Q]}$	$\oplus \text{r} \frac{\vdash \Gamma; \Pi, [Q]}{\vdash \Gamma; \Pi, [P \oplus Q]}$	no rule for 0	$! \frac{\vdash \Gamma; N}{\vdash \Gamma; [!N]}$	
Negative phase				
$\wp \frac{\vdash \Gamma; \Delta, N, M}{\vdash \Gamma; \Delta, N \wp M}$	$\perp \frac{\vdash \Gamma; \Delta}{\vdash \Gamma; \Delta, \perp}$	$\& \frac{\vdash \Gamma; \Delta, N \quad \vdash \Gamma; \Delta, M}{\vdash \Gamma; \Delta, N \& M}$	$\top \frac{}{\vdash \Gamma; \Delta, \top}$	$? \frac{\vdash \Gamma, P; \Delta}{\vdash \Gamma; \Delta, ?P}$

■ **Figure 2** LLKF: a two-zoned focused variant of cut-free LLK

$$\begin{array}{ll}
\Gamma ::= \cdot \mid \Gamma, P & \text{(Positive Sequent Contexts)} \\
\Delta ::= \cdot \mid \Delta, N & \text{(Negative Sequent Contexts)} \\
\Pi ::= \cdot \mid \Pi, \uparrow P \mid \Pi, \bar{a} & \text{(Reactive Sequent Contexts)}
\end{array}$$

LLKF proofs consist of alternating maximal *phases* based on the polarity of the principal formulas. These two phases are represented by two different sequent forms, given below. We follow Andreoli's original two-zoned (*dyadic*) convention for presenting the system because it is the most common style in presenting focused proof systems.

$$\begin{array}{ll}
\vdash \Gamma; \Delta & \text{(Negative Sequents)} \\
\vdash \Gamma; \Pi, [P] & \text{(Positive Sequents)}
\end{array}$$

The sequent $\vdash \Gamma; \cdot, [P]$ is abbreviated as $\vdash \Gamma; [P]$.

The focused rules of inference are shown in Figure 2. The most important rules are the *decision* rules wdc and dc that begin¹ a positive phase; in this phase, the focused formula (written inside $[]$) is principal, and the focus persists on the principal operands if they are of the same polarity. All essential choices are confined to this phase; they include: disjunctive choice (for \oplus), multiplicative choice (for \otimes), possible failures (for atoms, 1, and $!$, if the context is not of the correct form), and guaranteed failure (for 0, which has no rules). The positive phase switches to the negative phase with the rules for \downarrow or $!$. Observe that in the negative phase the rules can be applied in any order, and none of the negative rules can fail to apply. When no more negative rules can apply, a decision rule must be applied to restart the cycle. There are no structural rules of weakening or contraction because the rules treat the *unrestricted context* Γ as a set; in particular, contraction is part of the wdc rule.

The soundness of LLKF with respect to cut-free LLK is straightforward: forgetting the polarities, the focusing distinctions in sequents, and the rules $\{\text{dc}, \downarrow\}$; prefixing the elements of Γ with $?$ and using ct and wk to account for its additive treatment in the \otimes , 1 and id rules; and replacing wdc with the sequence ct then $?$ on the focused formula produces valid LLK proofs from LLKF proofs.

► **Notation 2.** Write $[P]$ (resp. $[N]$) for the unpolarized formula obtained from P (resp. N) by erasing all occurrences of \downarrow and \uparrow . Similarly we define $[\Delta]$.

¹ As usual, the intended reading of sequent rules is from conclusion to premises.

► **Theorem 3** (completeness of focusing). *If $\vdash [\Delta]$ in LLK, then $\vdash \cdot ; \Delta$ in LLKF.*

There are many ways to prove this theorem; we refer the interested reader to one of the standard approaches [13, 15]. One interesting feature of all such proofs is their unusual complexity forced by the rigidity of the focusing calculus. It is easier to show the completeness of focusing in the sequent calculus with more synthetic approaches [8].

3 Linear Logic in the Calculus of Structures

The calculus of structures is based on the observation that the connectives of linear logic preserve logical entailment, and therefore, any valid implication in the logic can be turned into a rewrite step on any subformula. Hence, there is no need to maintain a distinction between the connectives used in formulas, and structural meta-connectives such as the comma used to write sequents, or the meta conjunction among the premises of a binary rule. The original formulations of the calculus of structures [10, 11, 6] used *structures*, which are formulas modulo a syntactic congruence. We deviate from this tradition and use just the formulas, *i.e.*, we remove the syntactic congruence. Then, inference rules are allowed to operate on any subformula. These rules are therefore written in terms of *formula contexts* (ξ, ζ, \dots) , which are formulas with a single hole (written $\{ \}$), *i.e.*, they have the following grammar:

$$\xi, \zeta ::= \{ \} \mid A \star \xi \mid \xi \star A \mid !\xi \mid ?\xi \quad (\text{Formula Contexts})$$

where \star can stand for any binary connective (\otimes, \oplus, \wp , or $\&$). We write $\xi\{A\}$ for the formula formed by replacing the single occurrence of $\{ \}$ in ξ with the formula A . For example, if ξ is $!(a \& (\{ \} \otimes b)) \wp ?c$ and A is $\bar{a} \oplus b$, then $\xi\{A\}$ is $!(a \& ((\bar{a} \oplus b) \otimes b)) \wp ?c$.

A *derivation* \mathcal{D} in a system S with premise A and conclusion B is a rewriting path from A to B , using the rules in S . It is usually depicted as $s \parallel_{\mathcal{D}}^A B$. A *proof* \mathcal{P} in a system S , depicted as $s \parallel_{\mathcal{P}}^1 B$, is a derivation in S with premise 1.

3.1 The Unfocused Systems SLS and LS

The inference rules of the system SLS are given in Figure 3. The first two columns constitute the *multiplicative fragment*, the next two columns the *exponential fragment*, and the last two columns the *additive fragment*. The multiplicative and exponential fragment constitute system SELS, which is a variant of the system studied in [19]. The first four rows in Figure 3 constitute the *down fragment* of SLS, denoted by SLS_{\downarrow} , and the last four rows the *up fragment*, denoted by SLS_{\uparrow} . The down fragment corresponds to the cut-free version of the system, and following the tradition, we will call it LS. Each rule in either fragment is the dual of some rule in the other fragment, where the duals of a rule are formed by exchanging the premise and conclusion and negating both. Note that the two rules sl and sr (read *switch left* and *switch right*) are self-dual and therefore part of both fragments.² The rules ai_{\downarrow} and ai_{\uparrow} , called *atomic identity* and *atomic cut*, have the following general versions:

$$i_{\downarrow} \frac{\xi\{1\}}{\xi\{A \wp (A)^{\perp}\}} \quad i_{\uparrow} \frac{\xi\{(A)^{\perp} \otimes A\}}{\xi\{\perp\}}$$

² Note that our system SLS is slightly different from the presentation in [17, 18]. The reason for the differences is that we get stronger results (e.g., the down fragment does not need associativity for \otimes , \oplus , and $\&$), their proofs become simpler, and the relation to the focused systems is more evident.

$\text{ai}_\downarrow \frac{\xi\{1\}}{\xi\{a \wp \bar{a}\}}$	$\text{i}_\downarrow \frac{\xi\{1\}}{\xi\{!1\}}$	$\text{T}_\downarrow \frac{\xi\{1\}}{\xi\{\top\}}$	$\&\downarrow \frac{\xi\{1\}}{\xi\{1 \& 1\}}$
$\perp_\downarrow \frac{\xi\{A\}}{\xi\{A \wp \perp\}}$	$\text{com}_\downarrow \frac{\xi\{B \wp A\}}{\xi\{A \wp B\}}$	$\text{pr}_\downarrow \frac{\xi\{!(A \wp ?C)\}}{\xi\{!A \wp ?C\}}$	$?_\downarrow \frac{\xi\{A\}}{\xi\{?A\}}$
$\otimes_\downarrow \frac{\xi\{1\}}{\xi\{1 \otimes 1\}}$	$\text{asc}_\downarrow \frac{\xi\{A \wp (B \wp C)\}}{\xi\{(A \wp B) \wp C\}}$	$\text{ct}_\downarrow \frac{\xi\{?A \wp ?A\}}{\xi\{?A\}}$	$\text{wk}_\downarrow \frac{\xi\{\perp\}}{\xi\{?A\}}$
$\text{sl} \frac{\xi\{(A \wp C) \otimes B\}}{\xi\{(A \otimes B) \wp C\}}$	$\text{sr} \frac{\xi\{A \otimes (B \wp C)\}}{\xi\{(A \otimes B) \wp C\}}$	$\text{gc}_\downarrow \frac{\xi\{\top\}}{\xi\{\top \wp C\}}$	$\oplus_\downarrow \frac{\xi\{A\}}{\xi\{A \oplus B\}}$
$\otimes_\uparrow \frac{\xi\{\perp \wp \perp\}}{\xi\{\perp\}}$	$\text{asc}_\uparrow \frac{\xi\{(A \otimes B) \otimes C\}}{\xi\{A \otimes (B \otimes C)\}}$	$\text{ct}_\uparrow \frac{\xi\{!A\}}{\xi\{!A \otimes !A\}}$	$\text{wk}_\uparrow \frac{\xi\{!A\}}{\xi\{1\}}$
$\perp_\uparrow \frac{\xi\{A \otimes 1\}}{\xi\{A\}}$	$\text{com}_\uparrow \frac{\xi\{A \otimes B\}}{\xi\{B \otimes A\}}$	$\text{pr}_\uparrow \frac{\xi\{?A \otimes !C\}}{\xi\{?(A \otimes !C)\}}$	$?_\uparrow \frac{\xi\{!A\}}{\xi\{A\}}$
$\text{ai}_\uparrow \frac{\xi\{\bar{a} \otimes a\}}{\xi\{\perp\}}$	$\text{i}_\uparrow \frac{\xi\{?\perp\}}{\xi\{\perp\}}$	$\text{gc}_\uparrow \frac{\xi\{0 \otimes C\}}{\xi\{0\}}$	$\oplus_\uparrow \frac{\xi\{A \& B\}}{\xi\{B\}}$
		$\text{T}_\uparrow \frac{\xi\{0\}}{\xi\{\perp\}}$	$\&\uparrow \frac{\xi\{\perp \oplus \perp\}}{\xi\{\perp\}}$

■ **Figure 3** SLS, a symmetric calculus of structures for classical propositional linear logic. The fragment containing the first four rows is called LS.

Like in the sequent calculus, the general identity rule is derivable. By duality the same is true for the general cut rule. We have the following proposition, which is standard for systems in the calculus of structures (see. e.g., [17]).

► **Proposition 4.** *The rule i_\downarrow is derivable in SLS_\downarrow , and the rule i_\uparrow is derivable in SLS_\uparrow . Furthermore, every rule in SLS_\uparrow is derivable in $\text{SLS}_\downarrow + \text{i}_\uparrow$, and dually, every rule in SLS_\downarrow is derivable in $\text{SLS}_\uparrow + \text{i}_\downarrow$. ◀*

By an easy induction on the size of the proofs, one can show the following implications, expressing the relation to the sequent calculus.

► **Proposition 5.** *A formula A is provable in LLK with cut if and only if it is provable in SLS. And if A is provable in LLK without cut, then it is provable in LS. ◀*

We can now use Theorem 1 to show that provability in SLS implies provability in LS and that provability in LS, implies provability in LLK without cut.

► **Theorem 6** (cut elimination). *If a formula A is provable in SLS then it is provable in LS.*

► **Corollary 7.** *The rule i_\uparrow is admissible for LS. ◀*

The proof given in [17] for Theorem 6 relies on the sequent calculus and Theorem 1. In the following, we present a proof that is internal to SLS, *i.e.*, not using the sequent calculus. Due to lack of space, we can only give a sketch—all details can be found in [18]. First, observe that in any derivation \mathcal{D} in LS, all the instances of the contraction rule can be permuted to the bottom. This can be shown by an easy inductive argument.

► **Proposition 8.** *For every $\text{LS} \parallel_{\mathcal{D}}$ there is a B' such that $\text{LS} \setminus \{\text{ct}_\downarrow\} \parallel_{\mathcal{D}'}$ and $\{\text{ct}_\downarrow\} \parallel_{\mathcal{D}''}$. ◀*

For the internal cut-elimination proof of SLS, we will use a technique called *splitting*, first used in [10]. The central ingredients are Lemmas 10 – 12 and Lemma 14 below. Lemmas

10 – 12 say how the connectives behave in a *shallow* context, and Lemma 14 says how a general *deep* context can be reduced to a shallow one. For formally stating these lemmas, we need the notions of *linear killing context* and *killing context*, denoted by $\lambda\langle \ \rangle$ and $\kappa\langle \ \rangle$, respectively, and generated by this grammar:

$$\begin{aligned} \lambda &::= \top \mid \{ \} \mid \lambda \& \lambda \mid \lambda \otimes 1 \mid 1 \otimes \lambda && \text{(Linear Killing Contexts)} \\ \kappa &::= \top \mid \{ \} \mid \kappa \& \kappa \mid \kappa \otimes 1 \mid 1 \otimes \kappa \mid !\kappa && \text{(Killing Contexts)} \end{aligned}$$

We write $\lambda\langle \ \rangle^n$ (resp. $\kappa\langle \ \rangle^n$) to indicate that there are exactly n occurrences of $\{ \}$. Then, we write $\lambda\langle A_1, \dots, A_n \rangle$ (resp. $\kappa\langle A_1, \dots, A_n \rangle$) for the formula obtained from $\lambda\langle \ \rangle^n$ (resp. $\kappa\langle \ \rangle^n$) by replacing, from left to right, the n occurrences of $\{ \}$ by the formulas A_1, \dots, A_n . The two main properties of killing contexts are summarized in the following lemma.

► **Lemma 9.** *Let A, B_1, \dots, B_n , and $\lambda\langle \ \rangle^n$ and $\kappa\langle \ \rangle^n$ be given.*

1. *If B_1, \dots, B_n are provable in LS, then so are $\lambda\langle B_1, \dots, B_n \rangle$ and $\kappa\langle B_1, \dots, B_n \rangle$.*

2. *There are derivations*

$$\begin{array}{ccc} \lambda\langle A \wp B_1, \dots, A \wp B_n \rangle & \kappa\langle ?A \wp B_1, \dots, ?A \wp B_n \rangle \\ \text{LS} \parallel & \text{LS} \parallel \\ A \wp \lambda\langle B_1, \dots, B_n \rangle & ?A \wp \kappa\langle B_1, \dots, B_n \rangle \end{array} \quad \blacktriangleleft$$

We can now state the splitting lemmas.

► **Lemma 10** (binary splitting). *Let A, B , and K be formulas.*

1. *If $(A \& B) \wp K$ is provable in LS, then so are $A \wp K$ and $B \wp K$.*

2. *If $(A \oplus B) \wp K$ is provable in LS, then there is an $n \geq 0$ and K_1, \dots, K_n and $\lambda\langle \ \rangle^n$, such*

that

$$\begin{array}{ccc} \lambda\langle K_1, \dots, K_n \rangle & & \\ \text{LS} \parallel & \text{and for all } i \leq n \text{ we have} & \text{LS} \parallel \quad \text{or} \quad \text{LS} \parallel \\ K & A \wp K_i & B \wp K_i \end{array} \quad .$$

3. *If $(A \otimes B) \wp K$ is provable in LS, then there are $n \geq 0$ and $K_{A1}, K_{B1}, \dots, K_{An}, K_{Bn}$ and*

$\lambda\langle K_{A1} \wp K_{B1}, \dots, K_{An} \wp K_{Bn} \rangle$ and

$$\begin{array}{ccc} \lambda\langle \ \rangle^n, \text{ such that} & \text{LS} \parallel & \text{and} \quad \text{LS} \parallel \quad \text{and} \quad \text{LS} \parallel \\ & K & A \wp K_{Ai} \quad \text{and} \quad B \wp K_{Bi} \end{array}$$

for all $i \leq n$.

► **Lemma 11** (unit and atomic splitting). *Let x be an atom or a negated atom, and let K be a formula.*

4. *If $! \wp K$ is provable in LS, then there is a $\lambda\langle \ \rangle^n$ and a derivation*

$$\begin{array}{ccc} \lambda\langle \perp, \dots, \perp \rangle & & \\ \text{LS} \parallel & & \\ K & & \end{array} \quad .$$

5. *If $\perp \wp K$ is provable in LS, then so is K .*

6. *If $0 \wp K$ is provable in LS, then there is a $\lambda\langle \ \rangle^n$ and a derivation from $\lambda\langle \top, \dots, \top \rangle$ to K in LS.*

7. *If $x \wp K$ is provable in LS, then there is a $\lambda\langle \ \rangle^n$ and a derivation*

$$\begin{array}{ccc} \lambda\langle x^\perp, \dots, x^\perp \rangle & & \\ \text{LS} \parallel & & \\ K & & \end{array} \quad .$$

► **Lemma 12** (exponential splitting). *Let A and K be formulas.*

8. *If $!A \wp K$ is provable in LS, then there are $n \geq 0$ and K_1, \dots, K_n and $\lambda\langle \ \rangle^n$, such that*

$\lambda\langle K_1, \dots, K_n \rangle$ and for all $i \leq n$ we have

$$\begin{array}{ccc} \text{LS} \parallel & \text{with } K_i = \perp \text{ or } K_i = ?K_{i1} \wp \dots \wp ?K_{ih_i} \\ K & A \wp K_i \end{array}$$

for some $h_i \geq 1$.

9. *If $?A \wp K$ is provable in $\text{LS} \setminus \{\text{ct}_\perp\}$, then either K is provable in $\text{LS} \setminus \{\text{ct}_\perp\}$, or there are*

$n \geq 1$ and K_1, \dots, K_n and $\kappa\langle \ \rangle^n$, such that

$$\begin{array}{ccc} \kappa\langle K_1, \dots, K_n \rangle & & \\ \text{LS} \setminus \{\text{ct}_\perp\} \parallel & \text{and} \quad \text{LS} \setminus \{\text{ct}_\perp\} \parallel \\ K & A \wp K_i \end{array} \quad \text{for all } i \leq n.$$

All three splitting lemmas are proved in a similar way by an induction on the size of the given proof, and a case analysis on the bottommost rule instance. Although the statements of the splitting lemmas are different from the ones in [18], the proofs are almost literally the same. The purpose of the splitting lemmas is to prove the following lemma, which says that the rules of the up-fragment are admissible in a shallow context.

► **Lemma 13.** *Let K be a formula and let $r_{\uparrow} \frac{\xi\{F\}}{\xi\{G\}}$ be a rule in $\text{SLS}\uparrow$. If $F \wp K$ is provable in LS , then so is $G \wp K$.*

This is proved by using splitting to decompose the proof of $F \wp K$ into smaller pieces which can then be rearranged to build a proof of $G \wp K$. For the rules pr_{\uparrow} and $!_{\uparrow}$, we also need Proposition 8.

For proving Theorem 6, we need to extend Lemma 13 to general contexts. This is done by the *context reduction* lemma, whose proof is a straightforward induction on the context, repeatedly applying splitting.

► **Lemma 14 (context reduction).** *Let A be a formula and ξ be a context in which $\{ \}$ does not appear inside the scope of a $? \text{-modality}$. If $\xi\{A\}$ is provable in LS , then there exist an $n \geq 0$, a killing context $\kappa\langle \ \rangle$, and formulas K_{A_1}, \dots, K_{A_n} , such that*

$$\kappa\langle C \wp K_{A_1}, \dots, C \wp K_{A_n} \rangle \quad \text{for every formula } C \quad \text{and} \quad \frac{\text{LS} \parallel}{A \wp K_{A_i}} \quad \text{for every } i \leq n. \quad \blacktriangleleft$$

$$\frac{\text{LS} \parallel}{\xi\{C\}}$$

► **Lemma 15.** *Let $r_{\uparrow} \frac{\xi\{F\}}{\xi\{G\}}$ be a rule in $\text{SLS}\uparrow$ and let ξ be a context in which $\{ \}$ does not appear inside the scope of a $? \text{-modality}$. If $\xi\{F\}$ is provable in LS , then so is $\xi\{G\}$.*

This follows immediately from Lemma 13 and Lemma 14. In order to deal with $? \text{-contexts}$, we use the following lemma, proved by a simple rule permutation argument: any rule applied inside the scope of a $?$ can be permuted up until it leaves the scope of the $? \text{-modality}$.

► **Lemma 16 ($? \text{-reduction}$).** *For every proof \mathcal{P} in SLS there is a proof \mathcal{P}' in SLS with the same conclusion as \mathcal{P} , such that in \mathcal{P}' no inference rule is applied inside the scope of a $? \text{-modality}$.* ◀

Now Theorem 6 can be shown by first applying Lemma 16 and then eliminating all up-rules, starting with the topmost one, using Lemma 15.

3.2 LSF and LSS: Polarized, Focused, and Synthetic Variants of LS

In this section we study two complete polarized and focused variants of LS . Like in Andreoli's original formulation of focusing in the sequent calculus [1], we keep the general form of the rules of LS but modify them to respect polarity. The resulting calculus, called LSF , can be seen to be related to LS in the same way that LLKF is related to LLK . (In Section 4 below, we formalize the comparison between LLKF and LSF .) Just as in the sequent calculus, the proofs of completeness of the focusing restriction will become more manageable in a synthetic formulation of LSF , that we call LSS , which we present immediately after LSF .

Because LSF uses polarized formulas, the contexts in LSF are sensitive to the polarity of their holes. We use π and ρ for *positive formula contexts*, i.e., $\pi\{P\}$ is a well-formed polarized formula for any positive formula P . Likewise, we use ν and μ for *negative formula contexts*. Note that the polarity of $\pi\{P\}$ (resp. $\nu\{N\}$) need not itself be positive (resp. negative).

Decision			
$?_{\mathbb{F}} \frac{\nu\{\uparrow P\}}{\nu\{?P\}}$		$\text{int}_{\mathbb{F}} \frac{\nu\{\uparrow(P \oplus L)\}}{\nu\{\uparrow P \wp L\}}$	
Interaction			
$\text{aif}_{\mathbb{F}} \frac{\pi\{1\}}{\pi\{a \oplus \bar{a}\}}$	$\text{sl}_{\mathbb{F}} \frac{\pi\{(P \oplus L) \otimes Q\}}{\pi\{(P \otimes Q) \oplus L\}}$	$\text{sr}_{\mathbb{F}} \frac{\pi\{P \otimes (Q \oplus L)\}}{\pi\{(P \otimes Q) \oplus L\}}$	$\oplus_{\mathbb{F}} \frac{\pi\{P\}}{\pi\{P \oplus Q\}}$
	$\text{pc}_{\mathbb{F}} \frac{\pi\{\downarrow(N \wp L)\}}{\pi\{\downarrow N \oplus L\}}$	$\text{pr}_{\mathbb{F}} \frac{\pi\{!(N \wp ?P)\}}{\pi\{!N \oplus ?P\}}$	$\oplus_{\mathbb{F}} \frac{\pi\{Q\}}{\pi\{P \oplus Q\}}$
Superposition		Exponentiation	
$\text{dt}_{\mathbb{F}} \frac{\nu\{(M_1 \wp N) \& (M_2 \wp N)\}}{\nu\{(M_1 \& M_2) \wp N\}}$	$\text{gc}_{\mathbb{F}} \frac{\nu\{\top\}}{\nu\{\top \wp N\}}$	$\text{ct}_{\mathbb{F}} \frac{\nu\{?P \wp ?P\}}{\nu\{?P\}}$	$\text{wk}_{\mathbb{F}} \frac{\nu\{\perp\}}{\nu\{?P\}}$
Start		Congruence	
$\frac{\pi\{1\}}{\pi\{\downarrow 1\}}$	$\frac{\pi\{1\}}{\pi\{1 \otimes 1\}}$	$\frac{\pi\{1\}}{\pi\{!1\}}$	$\frac{\nu\{\uparrow 1\}}{\nu\{\uparrow 1 \& \uparrow 1\}}$
$\frac{\nu\{\uparrow 1\}}{\nu\{\top\}}$	$\frac{\nu\{N\}}{\nu\{N \wp \perp\}}$	$\frac{\nu\{N \wp M\}}{\nu\{M \wp N\}}$	$\frac{\nu\{(N_1 \wp N_2) \wp N_3\}}{\nu\{N_1 \wp (N_2 \wp N_3)\}}$

■ **Figure 4** LSF: a polarized and focused variant of LS

Inside positive contexts, we will use a notational device to mark the foci. To motivate this notation, consider the properties of the foci in LLKF sequents: they *interact* with the context by splitting it (for \otimes), by testing it for emptiness (for 1 and $!$), or by checking for the presence of negated atoms (for id). For LSF, these interactions need to be made incremental—formula by formula—because the formulas of the corresponding sequent context may not (yet) be present in a \wp relation at the point of the switch rules.

► **Definition 17.** An *interaction formula* (or simply an *interaction*) is a positive formula of the form $P \oplus L$. We call P in $P \oplus L$ the *focus* of the interaction, and L its *spine*.

Here, we use L to stand for *reactive formulas*, which correspond to the formulas that occur in LLKF sequents $\vdash \Gamma ; \Pi$, which are precisely those sequents that are introduced by the decision rules wdc and dc . Recall that such sequents represent formulas of the form $\wp(? \Gamma, \Pi)$. Reactive formulas, and their duals, the *active formulas*, therefore have this grammar:

$$\begin{aligned}
 R &::= a \mid !N \mid \downarrow N && \text{(Active Formulas)} \\
 L &::= \bar{a} \mid ?P \mid \uparrow P && \text{(Reactive Formulas)}
 \end{aligned}$$

Figure 4 lists the inference rules of LSF. A proof in LSF is a derivation with premise 1 or $\uparrow 1$. The start rules in Figure 4 define what it means to finish an LSF proof. The first start rule removes a pair of shifts from a 1 , and the other four are polarized versions of the rules \otimes_{\downarrow} , $!_{\downarrow}$, $\&_{\downarrow}$, and \top_{\downarrow} of LS (Figure 3). Interactions are created by the $\text{int}_{\mathbb{F}}$ rule, which corresponds to dc in LLKF. When the focus of the interaction involves a polarity shift, the interaction dissipates into an ordinary \wp using $\text{pc}_{\mathbb{F}}$, which is the analogue of the \downarrow rule of LLKF. In order to remain true to the spirit of LS, we keep contraction and decision as separate rules instead of building a specialized version of $\text{int}_{\mathbb{F}}$ that incorporates contraction. This lets us preserve the permutability of contraction (Proposition 8) even in the focused setting. To retain completeness, the $?_{\mathbb{F}}$ rule derelicts a $?$ to a \uparrow . The remaining rules for interactions follow the shape of the focus of the interaction, just as in the sequent calculus. For example, for \otimes , the rules $\text{sl}_{\mathbb{F}}$ and $\text{sr}_{\mathbb{F}}$ (that are the focused versions of the sl and sr rules of LS) send the spine of the interaction into one of the components of the focus. The remaining (non-interaction) rules are simply the direct polarity-respecting translations of the LS rules.

► **Theorem 18** (soundness). *For any N , if N is provable in LSF, then $\lfloor N \rfloor$ is provable in LS.*

Proof. Just replace $P \oplus L$ with $P \wp L$ and erase the polarity shifts. ◀

To show completeness, we will now move to a synthetic variant of LSF, called LSS, that keeps a sequence of interactions on subformulas of a focus together. While the correspondence with LLKF is clearer in LSF without this synthetic step, the proof of completeness is drastically simplified with synthetic formulations, a phenomenon that has also been observed for focusing in the sequent calculus [8].

The key observation needed to produce a synthetic variant of LSF is the following: in an interaction formula $P \oplus L$, the spine L is switched (using sl_F and sr_F) *deep* inside P until the focus of the interaction become active. During this switching, any \oplus -formulas in the focus are destructed by removing (using $\oplus|_F$ and $\oplus r_F$) one of its operands. Thus, we can define a special *tensor context*, written using π_\otimes and ρ_\otimes , with this grammar:

$$\pi_\otimes ::= \{ \} \mid \pi_\otimes \otimes P \mid P \otimes \pi_\otimes \quad (\text{Tensor Contexts})$$

Note that, because π_\otimes contains no shift or exponential connectives, any substitution $\pi_\otimes\{P\}$ is itself positive. Tensor contexts allow us to write the following synthetic forms of the interaction rules:

$$\text{sai}_F \frac{\nu\{\uparrow\pi_\otimes\{1\}\}}{\nu\{\uparrow\pi_\otimes\{a\} \wp \bar{a}\}} \quad \text{spc}_F \frac{\nu\{\uparrow\pi_\otimes\{\downarrow(N \wp L)\}\}}{\nu\{\uparrow\pi_\otimes\{\downarrow N\} \wp L\}} \quad \text{spr}_F \frac{\nu\{\uparrow\pi_\otimes\{!(N \wp ?P)\}\}}{\nu\{\uparrow\pi_\otimes\{!N\} \wp ?P\}}$$

► **Definition 19.** The system LSS is $\text{LSF} \setminus \{\text{int}_F, \text{ai}_F, \text{sl}_F, \text{sr}_F, \text{pc}_F, \text{pr}_F\} \cup \{\text{sai}_F, \text{spc}_F, \text{spr}_F\}$.

► **Theorem 20.** *Any formula is provable in LSS if and only if it is provable in LSF.*

Proof. Each instance of sai_F , spc_F , or spr_F can be derived by one of

$$\begin{array}{ccc} \text{ai}_F \frac{\nu\{\uparrow\pi_\otimes\{1\}\}}{\nu\{\uparrow\pi_\otimes\{a \oplus \bar{a}\}\}} & \text{pc}_F \frac{\nu\{\uparrow\pi_\otimes\{\downarrow(M \wp L)\}\}}{\nu\{\uparrow\pi_\otimes\{\downarrow M \oplus L\}\}} & \text{pr}_F \frac{\nu\{\uparrow\pi_\otimes\{!(M \wp ?P)\}\}}{\nu\{\uparrow\pi_\otimes\{!M \oplus ?P\}\}} \\ \frac{\{\text{sl}_F, \text{sr}_F\} \parallel \emptyset}{\nu\{\uparrow(\pi_\otimes\{a\} \oplus \bar{a})\}} & \frac{\{\text{sl}_F, \text{sr}_F\} \parallel \emptyset}{\nu\{\uparrow(\pi_\otimes\{\downarrow M\} \oplus L)\}} & \frac{\{\text{sl}_F, \text{sr}_F\} \parallel \emptyset}{\nu\{\uparrow(\pi_\otimes\{!M\} \oplus ?P)\}} \\ \text{int}_F \frac{\nu\{\uparrow\pi_\otimes\{a\} \oplus \bar{a}\}}{\nu\{\uparrow\pi_\otimes\{a\} \wp \bar{a}\}} & \text{int}_F \frac{\nu\{\uparrow(\pi_\otimes\{\downarrow M\} \oplus L)\}}{\nu\{\uparrow\pi_\otimes\{\downarrow M\} \wp L\}} & \text{int}_F \frac{\nu\{\uparrow(\pi_\otimes\{!M\} \oplus ?P)\}}{\nu\{\uparrow\pi_\otimes\{!M\} \wp ?P\}} \end{array}$$

where \emptyset can be constructed by a straightforward induction on π_\otimes .

In the other direction, note that \oplus is not commutative, *i.e.*, the order of the spines is fixed in iterations of \oplus . We can therefore permute any LSF proof to guarantee that the focus of any interaction isn't itself an interaction. Finally, we permute all instances of $\oplus|_F$ and $\oplus r_F$ as low as possible in the LSF proof so that all interactions are introduced by ai_F , pc_F or pr_F . The synthetic rules sai_F , spc_F and spr_F can now be easily recovered. ◀

For giving the proof of completeness of LSS with respect to LS, we proceed by inductive transformation of LS proofs in three steps. First, we rewrite the instances of the switch rules in LS to respect the restriction of the spines to reactive formulas, which corresponds to applying negative rules eagerly like in LLKF. Second, we use an auxiliary rule, called ps_F , that breaks the polarity restrictions by means of an extra pair of shifts in the premise. This rule allows us to transform any LS proof into a proof in $\text{LSS} \cup \{\text{ps}_F\}$. And third, we show that ps_F can be eliminated from $\text{LSS} \cup \{\text{ps}_F\}$.

For the first step, let LS' stand for LS where the rules sl and sr (see Figure 3) are replaced by the following rules sl' and sr' , respectively:

$$\text{sl}' \frac{\xi\{(A \wp [L]) \otimes B\}}{\xi\{(A \otimes B) \wp [L]\}} \quad \text{sr}' \frac{\xi\{A \otimes (B \wp [L])\}}{\xi\{(A \otimes B) \wp [L]\}}$$

Recall that L stands for reactive formulas. The following can be shown by an easy induction:

► **Lemma 21.** *The rules*

$$\text{cc}_\downarrow \frac{\xi\{(A \& B) \otimes C\}}{\xi\{(A \otimes C) \& (B \otimes C)\}} \quad \text{ga}_\downarrow \frac{\xi\{\top \otimes A\}}{\xi\{\top\}} \quad \bar{\perp}_\downarrow \frac{\xi\{A \wp \perp\}}{\xi\{A\}} \quad \bar{\text{dt}}_\downarrow \frac{\xi\{(A \& B) \wp C\}}{\xi\{A \wp C\} \& (B \wp C)} \quad \bar{\text{gc}}_\downarrow \frac{\xi\{\top \wp C\}}{\xi\{\top\}}$$

are admissible in LS^r . ◀

► **Lemma 22.** *If a formula A is provable in LS , then it is also provable in LS^r .*

Proof. Let the *size* of an instance of sl or sr with conclusion $\xi\{(A \otimes B) \wp C\}$ be defined as the number of symbols used in C . For transforming an LS proof into an LS^r proof we take two steps:

$$\text{LS} \parallel_{\mathcal{D}_1} \frac{}{A} \longrightarrow \text{LS}^r \cup \{\text{cc}_\downarrow, \text{ga}_\downarrow, \bar{\perp}_\downarrow, \bar{\text{dt}}_\downarrow, \bar{\text{gc}}_\downarrow\} \parallel_{\mathcal{D}_2} \frac{}{A} \xrightarrow{\text{Lemma 21}} \text{LS}^r \parallel_{\mathcal{D}_3} \frac{}{A}$$

For the first step, proceed by induction on the multi-set of the sizes of all switch instances in \mathcal{D}_1 , under multi-set ordering, showing that all instances of sl and sr can be reduced to sl^r and sr^r . Any instance of sl but not of sl^r can be replaced by one of the following derivations, reducing the size. (Note that we omitted some instances of asc_\downarrow and com_\downarrow .)

$$\begin{array}{ccc} \text{sl} \frac{(B \wp (G \wp H)) \otimes C}{((B \wp G) \otimes C) \wp H} & \bar{\perp}_\downarrow \frac{(B \wp \perp) \otimes C}{B \otimes C} & \bar{\text{dt}}_\downarrow \frac{(B \wp (G \& H)) \otimes C}{((B \wp G) \& (B \wp H)) \otimes C} \\ \text{sl} \frac{(B \otimes C) \wp (G \wp H)}{(B \otimes C) \wp \perp} & \perp_\downarrow \frac{B \otimes C}{(B \otimes C) \wp \perp} & \text{cc}_\downarrow \frac{((B \wp G) \otimes C) \& ((B \wp H) \otimes C)}{((B \wp G) \otimes C) \& ((B \otimes C) \wp H)} \\ & & \bar{\text{gc}}_\downarrow \frac{(B \wp \top) \otimes C}{\top \otimes C} \\ & & \text{ga}_\downarrow \frac{\top}{\top} \\ & & \text{gc}_\downarrow \frac{(B \otimes C) \wp \top}{(B \otimes C) \wp \top} \end{array}$$

The cases for sr are similar. ◀

The second step of the transformation of LS proofs to LSS proofs involves the following *partial switch* synthetic rule:

$$\text{ps}_F \frac{\nu\{\uparrow\pi_\otimes\{\downarrow(\uparrow P \wp L)\}\}}{\nu\{\uparrow\pi_\otimes\{P\} \wp L\}}$$

This rule has more shifts in the premise than in the conclusion and is therefore not derivable in LSS . It can permute above any rule in $\text{LSS} \setminus \{\text{sai}_F, \text{spc}_F, \text{spr}_F\}$. Before we can show that ps_F can always be eliminated from any proof of LSS , we need a lemma stating that $\downarrow\uparrow$ can be removed from an LSS proof with the use of ps_F .

► **Lemma 23.** *If there is a proof \mathcal{D} of a negative formula $\pi\{\downarrow\uparrow P\}$ in $\text{LSS} \cup \{\text{ps}_F\}$, then there is a proof of $\pi\{P\}$ in $\text{LSS} \cup \{\text{ps}_F\}$ of at most the same height as \mathcal{D} .*

Proof. Proceed by induction on the height of \mathcal{D} . In the base case, $\pi\{\downarrow\uparrow P\}$ is $\uparrow\downarrow\uparrow 1$ and the result is immediate. In the general case, consider the bottommost rule instance r in \mathcal{D} ; in most cases the induction hypothesis is directly applicable so the pair of shifts can simply be removed. The only interesting case is where r is a matching instance of spc_F . We replace it by an instance of ps_F as follows:

$$\text{spc}_F \frac{\nu\{\uparrow\pi_\otimes\{\downarrow(\uparrow P \wp L)\}\}}{\nu\{\uparrow\pi_\otimes\{\downarrow\uparrow P\} \wp L\}} \longrightarrow \text{ps}_F \frac{\nu\{\uparrow\pi_\otimes\{\downarrow(\uparrow P \wp L)\}\}}{\nu\{\uparrow\pi_\otimes\{P\} \wp L\}} \quad \blacktriangleleft$$

► **Remark.** Note that we also have the converse: if there is a proof of $\pi\{P\}$ in $\text{LSS} \cup \{\text{ps}_F\}$, then there is also a proof of $\pi\{\downarrow\uparrow P\}$ in $\text{LSS} \cup \{\text{ps}_F\}$.

► **Lemma 24.** *For every N , if $\lfloor N \rfloor$ is provable in LS^r , then N is provable in $\text{LSS} \cup \{\text{ps}_F\}$.*

Proof. Proceed by induction on the height of the LS^r proof \mathcal{D} of $\lfloor N \rfloor$ to build a proof of N in $\text{LSS} \cup \{\text{ps}_F\}$. The base case, where $\lfloor N \rfloor$ is 1, is trivial. Now make a case analysis for the bottommost rule instance r in \mathcal{D} . In most cases, we can simply replace r with the corresponding rule in LSS , and appeal to the induction hypothesis on the proof above r . The four interesting cases involve r being an instance of ai_\downarrow , sl^r , sr^r , or pr_\downarrow . We can apply the induction hypothesis to the proof above r and glue the result to one the following rule instances depending on the case:

$$\text{sai}_F \frac{\nu\{\uparrow 1\}}{\nu\{\uparrow a \wp \bar{a}\}} \quad \text{ps}_F \frac{\nu\{\uparrow(\downarrow(\uparrow P \wp L) \otimes Q)\}}{\nu\{\uparrow(P \otimes Q) \wp L\}} \quad \text{ps}_F \frac{\nu\{\uparrow(P \otimes \downarrow(\uparrow Q \wp L))\}}{\nu\{\uparrow(P \otimes Q) \wp L\}} \quad \text{spr}_F \frac{\nu\{\uparrow!(N \wp ?P)\}}{\nu\{\uparrow!N \wp ?P\}}$$

If the premises and conclusions do not match (because of extra $\downarrow\uparrow$ pairs) we appeal to Lemma 23 and the remark above. ◀

► **Lemma 25.** *The rule ps_F is height-preserving admissible in LSS .*

Proof. Given a proof \mathcal{D} of a negative formula N in $\text{LSS} \cup \{\text{ps}_F\}$, we prove by induction on the height of \mathcal{D} that there is a proof of N in LSS of at most the same height as \mathcal{D} . In the base case, N is $\uparrow 1$ and we are done. In the general case, we case-analyze the bottommost rule instance r in \mathcal{D} . If this is not an instance of ps_F , we appeal to the induction hypothesis on the proof above r and compose the result with r . In the case where r is an instance of ps_F , we consider the rule instance r_1 above r in \mathcal{D} , and consider the cases for r_1 . If r_1 is not a synthetic rule, then we can permute r up above r_1 and then appeal to the induction hypothesis on the proof now above r_1 . If r_1 was an instance of ct_F or dt_F , then we need to appeal to the induction hypothesis twice, which is possible because our reduction does not increase the height of \mathcal{D} . If r_1 was wk_F or gc_F , we do not need to appeal to the induction hypothesis. If $r_1 \in \{\text{sai}_F, \text{spc}_F, \text{spr}_F\}$, we merge r and r_1 by replacing them with a new instance of r_1 , as follows:

$$\begin{array}{ccc} \text{sai}_F \frac{\nu\{\uparrow\pi_\otimes\{\downarrow\uparrow\rho_\otimes\{1\}\}\}}{\nu\{\uparrow\pi_\otimes\{\downarrow(\uparrow\rho_\otimes\{a\} \wp \bar{a})\}\}} & \text{spc}_F \frac{\nu\{\uparrow\pi_\otimes\{\downarrow\uparrow\rho_\otimes\{\downarrow M \wp L\}\}\}}{\nu\{\uparrow\pi_\otimes\{\downarrow(\uparrow\rho_\otimes\{\downarrow M\} \wp L)\}\}} & \text{spr}_F \frac{\nu\{\uparrow\pi_\otimes\{\downarrow\uparrow\rho_\otimes\{!(M \wp ?L)\}\}\}}{\nu\{\uparrow\pi_\otimes\{\downarrow(\uparrow\rho_\otimes\{!M\} \wp ?P)\}\}} \\ \text{ps}_F \frac{\nu\{\uparrow\pi_\otimes\{\rho_\otimes\{a\}\} \wp \bar{a}\}}{\nu\{\uparrow\pi_\otimes\{\rho_\otimes\{a\}\} \wp \bar{a}\}} & \text{ps}_F \frac{\nu\{\uparrow\pi_\otimes\{\rho_\otimes\{\downarrow M\}\} \wp L\}}{\nu\{\uparrow\pi_\otimes\{\rho_\otimes\{\downarrow M\}\} \wp L\}} & \text{ps}_F \frac{\nu\{\uparrow\pi_\otimes\{\rho_\otimes\{!M\}\} \wp ?P\}}{\nu\{\uparrow\pi_\otimes\{\rho_\otimes\{!M\}\} \wp ?P\}} \\ \downarrow & \downarrow & \downarrow \\ \text{sai}_F \frac{\nu\{\uparrow\pi_\otimes\{\rho_\otimes\{1\}\}\}}{\nu\{\uparrow\pi_\otimes\{\rho_\otimes\{a\}\} \wp \bar{a}\}} & \text{spc}_F \frac{\nu\{\uparrow\pi_\otimes\{\rho_\otimes\{\downarrow M \wp L\}\}\}}{\nu\{\uparrow\pi_\otimes\{\rho_\otimes\{\downarrow M\}\} \wp L\}} & \text{spr}_F \frac{\nu\{\uparrow\pi_\otimes\{\rho_\otimes\{!(M \wp ?L)\}\}\}}{\nu\{\uparrow\pi_\otimes\{\rho_\otimes\{!M\}\} \wp ?P\}} \end{array}$$

Now we appeal to the induction hypothesis on the proof above r_1 to produce a new proof on which we apply Lemma 23 to get a proof \mathcal{D}' , with a conclusion matching the premise of the new instances resulting from the merge. We appeal to the induction hypothesis again on \mathcal{D}' and plug the result above the merged instance. Lastly, if r_1 is also an instance of ps_F , then we appeal to the induction hypothesis on the proof above r_1 and apply the technique used for the other cases. ◀

We now have all the ingredients for the completeness theorem for LSS .

► **Theorem 26.** *For any N , if $\lfloor N \rfloor$ is provable in LS , then N is provable in LSS .*

Proof. Let a proof of $\lfloor N \rfloor$ in LS be given. By Lemma 22, there is a proof of $\lfloor N \rfloor$ in LS^r . By Lemma 24, we have a proof of N in $\text{LSS} \cup \{\text{ps}_F\}$, and thus by Lemma 25 also in LSS . ◀

Note that since LSS and LSF are equivalent, Theorem 26 also proves the completeness of LSF with respect to LS .

4 Comparing Sequent and Structural Focusing

In order to justify the adjective “focused” for LSF, it is important to give a precise comparison with LLKF. In this section we shall prove that every LLKF proof can be simulated in LSF, and, conversely, every LSF proof has a corresponding LLKF proof. Both results are surprising, as there is no reason *a priori* that the two systems should have such a close correspondence. Indeed, there are significant differences such as the treatment of weakening and contraction and the incremental splitting of contexts around \otimes .

4.1 Simulating LLKF in LSF

First, let us simulate LLKF proofs in LSF, *i.e.*, show that LSF is adequate with respect to LLKF. The two proof systems are not isomorphic, so we use an abstraction.

► **Definition 27.** For a non-empty LLKF sequent σ and a polarized formula A , we say that A is a structural interpretation of σ , written $A \approx \sigma$, iff it can be derived from these rules:

$$\frac{}{P \approx (\vdash \cdot ; [P])} \quad \frac{Q \approx (\vdash \Gamma ; \Pi, [P])}{(Q \oplus L) \approx (\vdash \Gamma ; \Pi, L, [P])} \quad \frac{Q \approx (\vdash \Gamma, P ; \Pi, [P'])}{(Q \oplus ?P) \approx (\vdash \Gamma, P ; \Pi, [P'])} \quad \frac{Q \approx (\vdash \Gamma ; \Pi, [P'])}{\overline{Q} \approx (\vdash \Gamma, P ; \Pi, [P'])}$$

$$\frac{}{N \approx (\vdash \cdot ; N)} \quad \frac{M \approx (\vdash \Gamma ; \Delta)}{(M \wp N) \approx (\vdash \Gamma ; \Delta, N)} \quad \frac{M \approx (\vdash \Gamma, P ; \Delta)}{(M \wp ?P) \approx (\vdash \Gamma, P ; \Delta)} \quad \frac{M \approx (\vdash \Gamma ; \Delta)}{M \approx (\vdash \Gamma, P ; \Delta)}$$

In other words, structural interpretations can arbitrarily reorder the LLKF sequent and potentially erase or duplicate the unrestricted formulas, but they must preserve the multiplicities of the linear formulas. The simulation theorem shows that LSF can preserve the structural interpretations of each rule of LLKF.

- **Theorem 28 (simulation).** For any Γ, Δ, Π , and P ,
- If $\vdash \Gamma ; \Pi, [P]$ in LLKF, then there is a $Q \approx (\vdash \Gamma ; \Pi, [P])$ such that $\frac{1}{\uparrow 1} \frac{\text{LSF} \setminus \{\text{ct}_F\}}{Q}$.
 - If $\vdash \Gamma ; \Delta$ in LLKF, then there is a $N \approx (\vdash \Gamma ; \Delta)$ such that $\frac{\text{LSF} \setminus \{\text{ct}_F\}}{N}$.

Proof. By structural induction on the given LLKF proofs. ◀

► **Corollary 29 (completeness).** If $\vdash P_1, \dots, P_m ; N_1, \dots, N_n$ is provable in LLKF, then $N_1 \wp \dots \wp N_n \wp ?P_1 \wp \dots \wp ?P_m$ is provable in LSF.

Proof. We have:

$$\frac{\uparrow 1}{\frac{\text{Theorem 28}}{N_1 \wp \dots \wp N_n \wp^{u_1} ?P_1 \wp^{u_2} \dots \wp^{u_m} ?P_m}} \frac{\text{ct}_F, \text{wk}_F}{N_1 \wp \dots \wp N_n \wp ?P_1 \wp \dots \wp ?P_m}$$

where $M_1 \wp^u M_2$ stands for $M_1 \wp \underbrace{(M_2 \wp \dots \wp M_2)}_{u \text{ times}}$ if $u \geq 1$, and for M_1 if $u = 0$. ◀

4.2 Extracting LLKF Proofs from LSF Proofs

Let an LSF proof \mathcal{D} with conclusion N_0 be given. We present here an algorithm that extracts an LLKF proof of $\vdash \cdot ; N_0$ that is unique up to rule permutations which are entirely confined to the negative phases, *i.e.*, the extraction does not make any essentially non-deterministic choices. We begin by *labelling* the active and reactive formulas in N_0 , *i.e.*, we modify the grammar of formulas as follows:

$$\begin{aligned}
P, Q &::= a_u \mid !_u N \mid \downarrow_u N \mid P \otimes Q \mid 1 \mid P \oplus Q \mid 0 \mid P \oplus N \\
N, M &::= \bar{a}_u \mid ?_u P \mid \uparrow_u P \mid N \& M \mid \top \mid N \wp M \mid \perp
\end{aligned}$$

We use u, v, \dots for labels drawn from some infinite set, and Λ for a multi-set of labels. We write L_u or R_u to denote that the (re)active formula L or R has label u . The rules of LSF (Figure 4) are modified to be label-sensitive. The key cases are as follows:

$$\begin{aligned}
\text{int}_F & \frac{\nu\{\uparrow_u(P \oplus L_v)\}}{\nu\{\uparrow_u P \wp L_v\}} & \text{ai}_F & \frac{\pi\{1\}}{\pi\{a_u \oplus \bar{a}_v\}} & \text{pc}_F & \frac{\pi\{\downarrow_u(N \wp L_v)\}}{\pi\{\downarrow_u N \oplus L_v\}} & \text{pr}_F & \frac{\pi\{!_u(N \wp ?_v P)\}}{\pi\{!_u N \oplus ?_v P\}} \\
?_F & \frac{\nu\{\uparrow_u P\}}{\nu\{?_u P\}} & \text{ct}_F & \frac{\nu\{\uparrow_{u_1} P \wp \uparrow_{u_2} P\}}{\nu\{\uparrow_u P\}} \quad [\{u\} \rightsquigarrow \{u_1, u_2\}] & \text{wk}_F & \frac{\nu\{\perp\}}{\nu\{?_u P\}} \quad [\{u\} \rightsquigarrow \emptyset]
\end{aligned}$$

For all other rules the labelling is straightforward. The rules $\{\text{int}_F, \text{ai}_F, \text{pr}_F, \text{pc}_F\}$ in the first line above induce an ordering, written $<$, among the labels, with $u < v$ in each case. For ct_F , the labels u_1 and u_2 in the premise are assumed to be different from each other and from all labels in the conclusion of the rule. The rules ct_F and wk_F induce a rewrite relation \rightsquigarrow on multi-sets of labels that tracks the exponential uses of $?$ -formulas. We assume that if $u < v$ and $\{v\} \rightsquigarrow \Lambda, w$, then $u < w$. We label all active and reactive formulas in N_0 with unique labels and label every rule instance in \mathcal{D} as above. Note that the reflexive-transitive closure of this label ordering, written \leq , is a partial order.

Our algorithm will extract a labelled LLKF proof of $\vdash \cdot ; N_0$ where the unrestricted contexts contain positive formulas annotated with a multi-set of labels, *i.e.*, their elements will be of the form P_Λ . The wdc rule is modified to consume one of the available labels in this multi-set; if this multi-set is empty, then the rule is inapplicable. Likewise, the dc rule consumes the label of the linear reactive formula. Finally, in the $?$ rule, the label of the $?$ is normalized with respect to \rightsquigarrow .

$$\begin{aligned}
\text{wdc} & \frac{\vdash \Gamma, P_\Lambda ; \Pi, [P]}{\vdash \Gamma, P_{\Lambda, u} ; \Pi} & \text{dc} & \frac{\vdash \Gamma ; \Pi, [P]}{\vdash \Gamma ; \Pi, \uparrow_u P} & ? & \frac{(\{u\} \Downarrow \rightsquigarrow \Lambda) \quad \vdash \Gamma, P_\Lambda ; \Delta}{\vdash \Gamma ; \Delta, ?_u P}
\end{aligned}$$

The remaining rules of LLKF (Figure 2) can be modified to use labelled formulas in a straightforward manner. We say that a label is *available* in a labelled LLKF sequent if it is the label of some top-level formula in the sequent. The extraction of the labelled LLKF proof of $\vdash \cdot ; N_0$ proceeds by backwards proof search (*i.e.*, proof search from this goal sequent upwards by applying LLKF rules from conclusion to premises) with some constraints:

- For a negative sequent for which there are available negative rules (*i.e.*, rules in the negative phase in Figure 2), we apply one of these rules. The choice and order of the application of these rules is immaterial.
- From a negative sequent where no negative rules apply, we pick the unique \leq -smallest label from the available labels, and use wdc or dc as appropriate. Each pair of formulas in such a sequent has a corresponding pair of formulas in a \wp -relation in \mathcal{D} . Because we normalize with respect to \rightsquigarrow , every surviving available label in this sequent is involved in some instance of int_F with another available label of the sequent in \mathcal{D} . Thus, the available labels are \leq -connected, and, because there is a sub-proof in \mathcal{D} of the \wp -formula corresponding to this sequent, there is always a unique \leq -smallest label.
- For the \otimes rule of LLKF focused on $P \otimes Q$, we send those side formulas to the premise involving P whose labels are \leq -larger than some label of a subformula of P , and the rest to the premise involving Q . There is no splitting ambiguity: the sets of labels in subformulas of P and Q are disjoint because the labelling is unique.

- For \oplus , we repeat the same choices made in \mathcal{D} .

This algorithm is deterministic and always succeeds: it has no choice points that require backtracking. There are no labels in any subformula of $\mathbf{1}$, so no formula will ever be sent to a branch of a \otimes that has focus on $\mathbf{1}$, guaranteeing that its LLKF rule will succeed. Similarly, the sole formula that can be sent to a branch with focus on a_u will be a formula \bar{a}_v with $u < v$, and therefore the id rule of LLKF will succeed. Lastly, the only formulas that have labels \leq -greater than a $!$ -formula are $?$ -formulas, as ensured by the pr_F rule in LSF, so the corresponding $!$ rule of LLKF succeeds. The algorithm terminates because each decision rule consumes one of the finitely many labels of \mathcal{D} . We can erase the labels from the computed labelled LLKF proofs as a post-processing step.

References

- 1 J.-M. Andreoli. Logic programming with focusing proofs in linear logic. *J. of Logic and Computation*, 2(3):297–347, 1992.
- 2 A. Avron. The method of hypersequents in the proof theory of propositional non-classical logics. In *Logic: from foundations to applications: European logic colloquium*, pages 1–32. Clarendon Press, 1996.
- 3 D. Baelde and D. Miller. Least and greatest fixed points in linear logic. In *LPAR*, volume 4790 of *LNCS*, pages 92–106, 2007.
- 4 N. D. Belnap, Jr. Display logic. *J. of Philosophical Logic*, 11:375–417, 1982.
- 5 K. Brünnler. *Nested Sequents*. Habilitationsschrift, Universität Bern, 2010.
- 6 K. Brünnler and A. F. Tiu. A local system for classical logic. In *LPAR*, volume 2250 of *LNCS (LNAI)*, pages 347–361. Springer, 2001.
- 7 P. Bruscoli and A. Guglielmi. On the proof complexity of deep inference. *ACM Trans. on Computational Logic*, 10(2), 2009.
- 8 K. Chaudhuri. Focusing strategies in the sequent calculus of synthetic connectives. In *LPAR*, volume 5330 of *LNCS*, pages 467–481, 2008.
- 9 N. Guenot. Focused proof search for linear logic in the calculus of structures. In *ICLP*, volume 7 of *LIPICs*, pages 84–93. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010.
- 10 A. Guglielmi. A system of interaction and structure. *ACM Trans. on Computational Logic*, 8(1), 2007.
- 11 A. Guglielmi and L. Straßburger. Non-commutativity and MELL in the calculus of structures. In *CSL*, volume 2142 of *LNCS*, pages 54–68. Springer, 2001.
- 12 O. Laurent. *Étude de la Polarisation en Logique*. PhD thesis, Univ. Aix-Marseille II, 2002.
- 13 O. Laurent. A proof of the focalization property of linear logic. Unpublished note, 2004.
- 14 C. Liang and D. Miller. Focusing and polarization in linear, intuitionistic, and classical logics. *TCS*, 410(46):4747–4768, 2009.
- 15 D. Miller and A. Saurin. From proofs to focused proofs: a modular proof of focalization in linear logic. In *CSL*, volume 4646 of *LNCS*, pages 405–419. Springer, 2007.
- 16 P. O’Hearn and D. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.
- 17 L. Straßburger. A local system for linear logic. In *LPAR*, volume 2514 of *LNCS (LNAI)*, pages 388–402. Springer, 2002.
- 18 L. Straßburger. *Linear Logic and Noncommutativity in the Calculus of Structures*. PhD thesis, Technische Universität Dresden, 2003.
- 19 L. Straßburger. MELL in the Calculus of Structures. *TCS*, 309(1–3):213–285, 2003.
- 20 L. Straßburger and A. Guglielmi. A system of interaction and structure IV: The exponentials and decomposition, 2009. To appear in *ACM Trans. on Computational Logic*.