Focused Linear Logic and the λ-calculus

Taus Brock-Nannestad, Nicolas Guenot

Focused Linear Logic and the λ-calculus

Taus Brock-Nannestad[1,3]

INRIA Saclay & LIX, École Polytechnique

Nicolas Guenot[2,4]

IT University of Copenhagen

Abstract

Linear logic enjoys strong symmetries inherited from classical logic while providing a constructive framework comparable to intuitionistic logic. However, the computational interpretation of sequent calculus presentations of linear logic remains problematic, mostly because of the many rule permutations allowed in the sequent calculus. We address this problem by providing a simple interpretation of focused proofs, a complete subclass of linear sequent proofs known to have a much stronger structure than the standard sequent calculus for linear logic. Despite the classical setting, the interpretation relates proofs to a refined linear λ-calculus, and we investigate its properties and relation to other calculi, such as the usual λ-calculus, the $\lambda\mu$-calculus, and their variants based on sequent calculi.

Keywords: Linear Logic, Focusing, Lambda-calculus, Curry-Howard Correspondence

1 Introduction

The idea of *"proofs as programs"* found in the original Curry-Howard correspondence between intuitionistic natural deduction and the simply-typed λ-calculus has proved to be a powerful narrative, leading to the development of functional programming, with its expressive type systems and strong guarantees on the behaviour of programs. A number of variants and extensions have been proposed, such as the $\lambda\mu$-calculus [20], based on classical logic. In the midst of many developments around the computational interpretation of logical systems, linear logic [11] has taken an important place in this field by providing a refined view of logics and their computational meaning. However, it is mostly seen as a tool, the instrument of a methodology [3,14], rather than the object of a computational interpretation, at least in its standard form — its intuitionistic variant has been used to describe a linear λ-calculus [7]. The problem

[1] Email: taus.brock-nannestad@inria.fr

[2] Email: ngue@itu.dk

[3] Supported by the ERC Advanced Grant *ProofCert*.

[4] Supported by grant 10-092309 from the Danish Council for Strategic Research to the *Demtech* project.

lies in the framework of the sequent calculus in which it is described, which has a structure too lax to be conveniently represented through any kind of λ-terms. This fundamental problem has been tackled from the beginning using the formalism of proof-nets, described as *natural deduction for linear logic* [11], but this is a radical departure from traditional proof syntaxes, implying a use of graphs as representation, one way or another, to write programs. Keeping the standard syntax of proof trees requires changing the kind of programs considered, and interpretations based on pattern-matching [22] or processes [1,6,23] have been proposed.

In another setting, *focusing* [4] has been developed to improve proof search in linear logic by identifying a complete subset of proofs endowed with a strong structure. As this normal form was studied in details, it became apparent that the notion of *polarity* and the associated permutability properties leading to focused systems was an essential aspect of linear proof theory [15]. However, the focused sequent calculus for linear logic, just as its unfocused variant, has not appeared as a framework of choice for a direct computational interpretation in the Curry-Howard tradition. In this paper, we propose a simple interpretation of the most standard focused presentation of **MELL**, showing how the structure of focused proofs can be described by refined linear λ-terms as found in an intuitionistic setting, despite the *"classical"* nature of this logic — and in particular, despite the presence of the $\parr$ connective.

The key to this interpretation is the use of an explicitly polarised syntax, where the negative formulas type computations while positive formulas type values, as done in the *call-by-push-value* framework [17]. Moreover, we consider a strongly focused system, where inversion is performed maximally and in an ordered fashion, thus yielding normal forms where no two inference rule instances can be permuted, although entire focused phases can still be permuted. The extracted calculus, that we call $\lambda\overline{\pi}$, has no explicit control operator, but its type system allows the encoding of calculi with control, such as $\lambda\mu$, through a relatively simple translation. We present in Section 2 our focused proof system for **MELL** along with a term assignment, and discuss its basic properties.

The two fundamental results, cut elimination and focalisation, are proved in Section 3, and we discuss the computational interpretation of these two theorems, when viewing their proofs as transformations of terms in the $\lambda\overline{\pi}$-calculus. The reduction of cuts, done in big steps, corresponds to the expected notion of reduction, based on substitution and the decomposition of pairs. Focusing corresponds to a reorganisation of terms that simplifies the structure of a term by rearranging the position of its subterms, merging values previously kept separated by unrelated phases of computation.

Finally, we discuss in Section 4 the expressivity of the $\lambda\overline{\pi}$-calculus by considering fragments and encodings of known calculi into these fragments. Of particular interest are the sequent calculus variants of the simply-typed λ-calculus and of the $\lambda\mu$-calculus, which are closely related to their originals stemming from natural deduction. Note that $\lambda\overline{\pi}$ has a rich structure: it is a sequent-based variant of $\lambda\mu$ with a notion of linearity, in which terms can be applied to trees rather than just lists of arguments. We conclude in Section 5 and discuss further investigations, from generalisations to richer logics to practical applicability.

Related work. As mentioned, the computational meaning of the standard system for *"classical"* linear logic has not been investigated as much as the interpretations given for intuitionistic logic or classical logic. A detailed description of focusing in intuitionistic logic, with proof terms, can for example be found in [21]. In the classical setting, the study of computation in the sequent calculus [9] has lead to a system called **L** providing a syntax extending the $\lambda\mu$-calculus in a symmetric way, and this system has been studied in the linear setting [19]. However, this system is not focused in general and cuts, performing the selection of a formula to focus on, cannot all be eliminated. Also related is the work on *polarised linear logic* [15], and in particular the encoding the $\lambda\mu$-calculus in polarised proof-nets [16]. We discuss this connection in Section 4.

2 Focused Proofs and Linear λ-terms

Focusing can be seen as a way of *structuring* proofs. Take for instance the standard rules of the multiplicative fragment of linear logic:

$$\frac{}{\vdash a, \overline{a}} \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B} \qquad \frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}$$

Proofs within this fragment have very little structure beyond that enforced by the subformula relation. Focusing allows us to enforce further structure in two ways. Firstly, the rule introducing the $\parr$ connective is invertible (i.e. the conclusion implies the premise), hence we can assume that this inversion property is always applied maximally within a proof. In other words, no $\otimes$ is decomposed if there is a $\parr$ in the remaining context. Secondly, and perhaps most importantly, the decomposition of $\otimes$ can be rearranged into *maximal chains* of $\otimes$-decompositions. Note that the $\otimes$ rule is *not* invertible, as it requires the linear context to be split into two parts, and this split may not be known beforehand. Thus, decomposing, say, $A \otimes (B \otimes C)$ results in subderivations containing A and $B \otimes C$ respectively. In the focusing discipline, we would require that $B \otimes C$ was decomposed immediately as well.

We enforce the maximality of the inversion (or asynchronous) and chaining (or synchronous) phases by dividing the sequent into two parts. During inversion, we maintain a list of potentially invertible formulas, and always decompose the first element of this list. If the top connective is a $\parr$, we put both subformulas back into the list, otherwise we move the formula into the other part of the sequent. In this way, we ensure that every $\parr$ formula in the context gets decomposed, if possible. During the chaining phase, we maintain a *stoup* containing a single formula, called the *"focus"*. When decomposing a $\otimes$, the two subformulas are put in this stoup in the premises, thus ensuring that these formulas are decomposed in turn, if possible.

We consider the multiplicative-exponential fragment of linear logic [11] with the purely linear connectives $\parr$ and $\otimes$, the exponential modalities ? and ! as well as the explicit polarity shifts $\uparrow$ and $\downarrow$ — as linear mediation between positives and negatives [15]. We also assume given a countable set $\mathcal{A}$ of atoms partitioned such that any atom a has a uniquely defined negative counterpart $\overline{a}$ in $\mathcal{A}$. The grammar of formulas in this polarised variant of **MELL** is divided in two classes:

$$P, Q \ ::= \ \downarrow\overline{a} \mid \downarrow N \mid P \otimes Q \mid !N \qquad\qquad N, M \ ::= \ \uparrow a \mid \uparrow P \mid N \parr M \mid ?P$$

$$\frac{}{x \vdash \Psi; x : \uparrow a \Downarrow \downarrow \overline{a}} \qquad \frac{t \vdash \Psi; \Gamma \Uparrow N}{\lfloor t \rfloor \vdash \Psi; \Gamma \Downarrow \downarrow N} \qquad \frac{t \vdash \Psi; \Gamma, x : \uparrow A \Uparrow S}{\lambda x.t \vdash \Psi; \Gamma \Uparrow \uparrow A, S}$$

$$\frac{t \vdash \Psi; \Gamma \Uparrow N, S \quad p \vdash \Psi; \Delta \Downarrow N^{\perp}}{t\, p \vdash \Psi; \Gamma, \Delta \Uparrow S} \qquad \frac{p \vdash \Psi; \Gamma \Downarrow P}{x\, p \vdash \Psi; \Gamma, x : \uparrow P \Uparrow \cdot}$$

$$\frac{t \vdash \Psi; \Gamma \Uparrow N, M, S}{\pi.t \vdash \Psi; \Gamma \Uparrow N \parr M, S} \qquad \frac{p \vdash \Psi; \Gamma \Downarrow P \quad q \vdash \Psi; \Delta \Downarrow Q}{(p, q) \vdash \Psi; \Gamma, \Delta \Downarrow P \otimes Q}$$

$$\frac{t \vdash \Psi, x : ?P; \Gamma \Uparrow S}{\lambda !x.t \vdash \Psi; \Gamma \Uparrow ?P, S} \qquad \frac{t \vdash \Psi; \cdot \Uparrow N}{!t \vdash \Psi; \cdot \Downarrow !N} \qquad \frac{p \vdash \Psi, x : ?P; \Gamma \Downarrow P}{!x\, p \vdash \Psi, x : ?P; \Gamma \Uparrow \cdot}$$

Fig. 1. Inference rules for focused **MELL** with $\lambda\overline{\pi}$-terms

where P and N denote positive and negative formulas respectively. The notion of duality $(\cdot)^{\perp}$ extending the relation between a and $\overline{a}$ to all formulas is defined as usual in linear logic. Notice that in our syntax, atoms always appear immediately under a polarity shift: this is an explication of the *bias*, the choice of the polarity of a given atom. A polarised formula therefore contains all the required polarity information. In the following, we will write $\uparrow A$ to denote either $\uparrow a$ or $\uparrow P$, and $\downarrow A$ for either $\downarrow \overline{a}$ or $\downarrow N$. The sequent calculus shown above in Figure 1 is the standard *triadic* form of the focused system [4] for **MELL**, made more precise by the use of polarity shifts. It operates on two kinds of sequents:

$$\begin{cases} \vdash \Psi; \Gamma \Uparrow S & : \quad \textit{asynchronous} \text{ sequent, where } S \text{ is a sequence of negatives} \\ \vdash \Psi; \Gamma \Downarrow P & : \quad \textit{synchronous} \text{ sequent, where } P \text{ is a single positive} \end{cases}$$

where Ψ and Γ are multisets containing named formulas, written $x : ?P$ and $x : \uparrow A$ respectively. As usual, we assume that the variables affixed to these formulas are distinct. The list S can be empty, and represents the part of the context treated in the inversion phase, while P in the other sequent is the positive decomposed in a focus phase — more details on this system can be found in [4]. Note that formulas in Ψ are subject to weakenings and contractions, since they are exponential.

The sequent calculus in Figure 1 comes with a term assignment: it is a variant of the linear λ-calculus separating terms from values. This distinction is not surprising given the known impact of focusing on computation [10], but the direct typing of λ-terms by proofs of the *"classical"* flavour of linear logic is. Indeed, although intuitionistic linear logic has been interpreted through linear λ-terms [7], the standard linear logic using $\parr$ in a sequent calculus has been connected to processes [6,23] or pattern-matching [22] only. This is an illustration of the ability of focalisation to shed light on the computational meaning of proofs in the sequent calculus.

We call the language used to represent focused **MELL** proofs the $\lambda\overline{\pi}$-calculus: it can be viewed as a linear variation on the $\overline{\lambda}$-calculus representing **LJT** proofs proposed by Herbelin [12]. However, it uses pairs and an unpairing operator π rather than lists of arguments. The grammar of *terms* and *values* is:

$$t, u \;::=\; \lambda x.t \mid x\,p \mid t\,p \mid \pi.t \mid \lambda !x.t \mid !x\,p \qquad p, q \;::=\; x \mid \lfloor t \rfloor \mid (p, q) \mid !t$$

where t and p denote a term and a value respectively. Beyond having constructs for abstraction and applications, the $\lambda\overline{\pi}$-calculus features *thunks* $\lfloor t \rfloor$ turning a term into a value, as found for example in *call-by-push-value* [17]. In addition, because the core language is linear, $\lambda\overline{\pi}$ has replicating variants of the abstraction, variable application, and thunk constructs. The relation between proofs and terms is tight: there is one-to-one correspondence in the fragment covered by **MELL**. This strong connection between the proofs of a highly symmetric logic and a relatively standard λ-calculus is made possible by the additional structure obtained through focusing.

Identity expansion. As usual in linear logic, the more general form of the identity axiom applicable on a compound formula is admissible in this system. However, due to the polarised setting, this requires a precise definition of the *expansion* of a negative formula N, denoted by N^* and defined as follows:

$$(\uparrow A)^* \;=\; \uparrow A \qquad (N \parr M)^* \;=\; N^*, M^* \qquad (?P)^* \;=\; ?P$$

so that we can relate any N to the pair of a persistent and a linear context (Ψ, Γ) through expansion. In the following, we write $\lceil \Gamma \rceil$ for the set of formulas contained in Γ. We can now prove in a mutual induction the following two lemmas justifying the generalisation of the identity axiom.

Lemma 2.1 *The sequents* $\vdash \Psi; x : \uparrow P \Uparrow P^\perp$ *and* $\vdash \Psi, x : ?P; \cdot \Uparrow P^\perp$ *are provable.*

Proof. In the first case, we can decompose $P^\perp$ eagerly until we obtain a premise of the shape $\vdash \Psi, \Omega; \Gamma, x : \uparrow P \Uparrow \cdot$ and focus on P, so that we conclude by Lemma 2.2. The other case is treated the same way, but uses the exponential focus rule. □

Lemma 2.2 (Identity expansion) *If* $N^* = \lceil \Psi, \Gamma \rceil$ *then we have* $\vdash \Psi, \Omega; \Gamma \Downarrow N^\perp$.

Proof. We proceed by induction on the formula N. In the base case, N is of the shape $\uparrow a$ and we conclude using the axiom rule. In the general case, if N is of the shape $M \parr L$ we apply the induction hypothesis on M and L separately and compose the proofs obtained using the $\otimes$ rule. If N is of the shape $\uparrow P$ or $?P$ then we conclude using Lemma 2.1 on $P^\perp$ and either the $\downarrow$ rule or the $!$ rule. □

An immediate consequence of this last lemma is that the identity axiom can now be generalised into the following two rules:

$$\frac{}{x \vdash \Psi; x : \uparrow P \Downarrow \downarrow P^\perp} \qquad\qquad \frac{}{!x \vdash \Psi, x : ?P; \cdot \Downarrow !P^\perp} \tag{1}$$

and if such rules are added, we can type terms that are essentially not η-long. Moreover, the proof of identity expansion provides us with a transformation of terms comparable to η-expansion, where a variable value is replaced with a thunk of a more complex term depending on its type in the unordered contexts. The rules are:

$$\begin{array}{llll}
x \to_\eta \lfloor \lambda z.(x\ z) \rfloor & \textit{if } x : \uparrow\!\downarrow N & !x \to_\eta !\lambda z.(!x\ z) & \textit{if } x : ?\!\downarrow N \\
x \to_\eta \lfloor \pi.\lambda y.\lambda z.x\ (y, z) \rfloor & \textit{if } x : \uparrow(P \otimes Q) & !x \to_\eta !\pi.\lambda y.\lambda z.!x\ (y, z) & \textit{if } x : ?(P \otimes Q) \\
x \to_\eta \lfloor \lambda !z.(x\ !z) \rfloor & \textit{if } x : \uparrow !N & !x \to_\eta !\lambda !z.(!x\ !z) & \textit{if } x : ?!N
\end{array}$$

when decomposing the result of Lemma 2.2 into small steps. In the following, we will only consider terms in η-long normal form — that is, such that none of the rules above apply and therefore corresponding to a proof with atomic identity axioms.

3 Cut Elimination and Focusing as Transformations

We now turn to the dynamics of the system, by considering proof transformations and their effect on corresponding terms. As usual, cut elimination is interpreted as a rewrite system implementing computation, and in our system, the focalisation result is interpreted as well, corresponding to a reorganisation that simplifies $\lambda\overline{\pi}$-terms.

Cut elimination in focused sequent calculi is usually presented as the admissibility of a collection of cut rules. A step of cut reduction consists in having the cut interact with the rules appearing directly above it, either permuting the cuts above said rules, or decomposing the cut into smaller instances. This presentation has several drawbacks, however. Firstly, the *"commutative cuts"* have no computational meaning, and serve only to permute inference rules around until the next step of computation can take place. Moreover, this small-step nature of the reduction almost invariably leads to a failure of strong normalisation for the associated system of reductions.

Since we are interested in the dynamics of cut elimination, we will opt for a very conservative view. We propose that there is only one relevant instance of the cut rule, the one shown in Figure 1:

$$\frac{\vdash \Psi; \Gamma \Uparrow N, S \quad \vdash \Psi; \Delta \Downarrow N^{\perp}}{\vdash \Psi; \Gamma, \Delta \Uparrow S}$$

Note that because the calculus enforces maximal synchronous and asynchronous phases, the formulas N and $N^{\perp}$ are both principal in the premises. In other words, this cut rule represents exactly the *principal* cases of the cut-elimination argument. To handle the cases where the principal formula is not decomposed, we see the derivation with the non-principal formula as the composition of a *context* and a subderivation that is again principal. For the sake of readability, we elide the proof terms for the time being: later in this section, we will show the set of reductions that form the computational content of the following theorems. To ease the notation, we use $\vdash \Psi; \Gamma \Updownarrow \Sigma$ as a shorthand for either $\vdash \Psi; \Gamma \Downarrow P$ or $\vdash \Psi; \Gamma \Uparrow S$.

The first step is to prove a series of decomposition lemmas based on a case analysis of the structure of derivations. We omit the proof of the most basic lemma, concerning formulas of the shape $\uparrow a$.

Lemma 3.1 (Atomic decomposition) *For any proof $\mathcal{E} :: \vdash \Psi; \Gamma, x : \uparrow a \Updownarrow \Sigma$ there exists an open derivation $\mathcal{G}$ from $\vdash \Psi; x : \uparrow a \Downarrow \downarrow \overline{a}$ to $\vdash \Psi; \Gamma, x : \uparrow a \Updownarrow \Sigma$.*

Lemma 3.2 (Linear decomposition) *Any derivation $\mathcal{E} :: \vdash \Psi; \Gamma, x : \uparrow P \Updownarrow \Sigma$ can be decomposed into a derivation $\mathcal{F} :: \vdash \Psi, \Omega; \Delta \Downarrow P$ and an open derivation $\mathcal{G}$ from $\vdash \Psi, \Omega; \Delta \Uparrow \cdot$ to $\vdash \Psi; \Gamma \Updownarrow \Sigma$, and this open derivation does not contain an instance of the ! rule between the open hypothesis and the conclusion.*

Proof. The proof proceeds by induction on the derivation $\mathcal{E}$. If $\mathcal{E}$ ends with a rule that does not focus on $x : \uparrow P$ in its premise, we place this rule on top of the open derivation $\mathcal{G}$. If the rule focuses on $x : \uparrow P$, then this is the desired derivation $\mathcal{F}$, and we are done. At no point do we encounter the ! rule in $\mathcal{E}$, as this rule requires an empty linear context, which is impossible due to the presence of $x : \uparrow P$. □

The requirement that $\mathcal{G}$ does not contain instances of the ! rule is important, as it justifies extending the linear context uniformly across the open derivation: If $\mathcal{G}$ is an open derivation from $\vdash \Psi, \Omega; \Delta \Uparrow \cdot$ to $\vdash \Psi; \Gamma \Updownarrow \Sigma$ satisfying the requirement, then it is also an open derivation from $\vdash \Psi, \Omega; \Delta, \Phi \Uparrow \cdot$ to $\vdash \Psi; \Gamma, \Phi \Updownarrow \Sigma$.

When the formula is of the shape $?P$ and appears under the name x in the persistent context, we define the *multiplicity* $|x|_\mathcal{E}$ of x in a derivation $\mathcal{E}$ as the number of times $x : ?P$ is principal in a focus rule in $\mathcal{E}$ — how many focus phases are started on x. This requires considering proofs *modulo* the usual notion of α-equivalence.

Lemma 3.3 (Persistent decomposition) *For any proof $\mathcal{E} :: \vdash \Psi, x : ?P; \Gamma \Updownarrow \Sigma$, either we have $\mathcal{E} :: \vdash \Psi; \Gamma \Updownarrow \Sigma$ or there exists a derivation $\mathcal{F} :: \vdash \Psi, \Omega; \Delta \Downarrow P$ and an open derivation $\mathcal{G}$ from $\vdash \Psi, \Omega; \Delta \Uparrow \cdot$ to $\vdash \Psi, x : ?P; \Gamma \Updownarrow \Sigma$, such that $|x|_\mathcal{G} < |x|_\mathcal{E}$.*

Proof. Again, we prove this by induction on the given derivation $\mathcal{E}$. Intuitively, we pick a topmost focus rule using $x : ?P$, yielding the derivation $\mathcal{F}$. The remaining derivation, in which x necessarily has a lower multiplicity, then becomes $\mathcal{G}$. □

On the basis of these decomposition lemmas, we can complete our cut elimination argument. We state it here simply as a weak normalisation argument, although we expect the calculus to admit a strong normalisation argument as well, given its similarity to the *linear substitution calculus* [2].

Theorem 3.4 (Cut elimination) *The following two cut principles hold in focused **MELL**, assuming the derivations $\mathcal{D}$ and $\mathcal{E}$ are cut-free:*

(i) if $\mathcal{D} :: \vdash \Psi; \Gamma \Uparrow P^\perp, S$ and $\mathcal{E} :: \vdash \Psi; \Delta \Downarrow P$, there exists $\mathcal{F} :: \vdash \Psi; \Gamma, \Delta \Uparrow S$,
(ii) if $\mathcal{D} :: \vdash \Psi, x : ?P; \Gamma \Uparrow S$ and $\mathcal{E} :: \vdash \Psi; \cdot \Uparrow P^\perp$, there exists $\mathcal{F} :: \vdash \Psi; \Gamma \Uparrow S$.

Proof. The first statement is established by induction on the structure of P. Note that by design, both P and $P^\perp$ must be principal in the assumptions, hence the structure of the formula also forces what the final rule must be in the $\mathcal{D}$ and $\mathcal{E}$ derivations.

- If $P = Q \otimes R$:

$$\dfrac{\overset{\mathcal{D}}{\vdash \Psi; \Gamma \Uparrow Q^\perp, R^\perp, S}}{\vdash \Psi; \Gamma \Uparrow Q^\perp \parr R^\perp, S} \qquad \dfrac{\overset{\mathcal{E}_1}{\vdash \Psi; \Delta_1 \Downarrow Q} \quad \overset{\mathcal{E}_2}{\vdash \Psi; \Delta_2 \Downarrow R}}{\vdash \Psi; \Delta_1, \Delta_2 \Downarrow Q \otimes R}$$

  we need to construct a derivation $\mathcal{F} :: \vdash \Psi; \Gamma, \Delta_1, \Delta_2 \Uparrow S$. We proceed by applying the induction hypothesis on $\mathcal{D}$ and $\mathcal{E}_1$ to obtain $\mathcal{D}' :: \vdash \Psi; \Gamma, \Delta_1 \Uparrow R^\perp, S$, and then apply the induction hypothesis again on $\mathcal{D}'$ and $\mathcal{E}_2$ to produce the desired derivation for $\vdash \Psi; \Gamma, \Delta_1, \Delta_2 \Uparrow S$.

- If $P = \downarrow\overline{a}$:

$$\dfrac{\overset{\mathcal{D}}{\vdash \Psi; \Gamma, x : \uparrow a \Uparrow S}}{\vdash \Psi; \Gamma \Uparrow \uparrow a, S} \qquad \mathcal{E} :: \dfrac{}{\vdash \Psi; y : \uparrow a \Downarrow \downarrow\overline{a}}$$

where the second derivation is forced since the grammar disallows atoms without a polarity shift. We need to construct a derivation $\mathcal{F} :: \vdash \Psi; \Gamma, y : \uparrow a \Uparrow S$: if we were only interested in provability, we could simply reuse $\mathcal{D}$. In order to get the term reductions to behave as expected, we instead reason as follows: by applying Lemma 3.1 on $\mathcal{D}$ we obtain an open derivation $\mathcal{G}$ from $\vdash \Psi, \Omega; x : \uparrow a \Downarrow \downarrow\overline{a}$ to $\vdash \Psi; \Gamma, x : \uparrow a \Uparrow S$ that we compose with $\mathcal{E}$ to produce the desired result.

- If $P = \downarrow Q^{\perp}$:

$$\dfrac{\begin{array}{c}\mathcal{D}\\ \vdash \Psi; \Gamma, x : \uparrow Q \Uparrow S\end{array}}{\vdash \Psi; \Gamma \Uparrow \uparrow Q, S} \qquad \dfrac{\begin{array}{c}\mathcal{E}\\ \vdash \Psi; \Delta \Uparrow Q^{\perp}\end{array}}{\vdash \Psi; \Delta \Downarrow \downarrow Q^{\perp}}$$

  we need to construct a derivation $\mathcal{F} :: \vdash \Psi; \Gamma, \Delta \Uparrow S$. By applying Lemma 3.2 on $\mathcal{D}$, we obtain a derivation $\mathcal{D}' :: \vdash \Psi, \Omega; \Gamma' \Downarrow Q$ and an open derivation $\mathcal{G}$ from $\vdash \Psi, \Omega; \Gamma' \Uparrow \cdot$ to $\vdash \Psi; \Gamma \Uparrow S$. Then by the induction hypothesis on $\mathcal{D}'$ and $\mathcal{E}$ we produce a derivation of $\vdash \Psi, \Omega; \Gamma', \Delta \Uparrow \cdot$ and compose it with $\mathcal{G}$ to obtain a derivation of $\vdash \Psi; \Gamma, \Delta \Uparrow S$. Notice that the linear context has changed from Γ' to Γ', Δ, but this extension of the linear context is done uniformly across the open derivation, and hence is not a problem.

- If $P = !Q^{\perp}$:

$$\dfrac{\begin{array}{c}\mathcal{D}\\ \vdash \Psi, x : ?Q; \Gamma \Uparrow S\end{array}}{\vdash \Psi; \Gamma \Uparrow ?Q, S} \qquad \dfrac{\begin{array}{c}\mathcal{E}\\ \vdash \Psi; \cdot \Uparrow Q^{\perp}\end{array}}{\vdash \Psi; \cdot \Downarrow !Q^{\perp}}$$

  we can directly apply the second induction hypothesis on $\mathcal{D}$ and $\mathcal{E}$.

For the second statement, we proceed by induction on the multiplicity $|x|_{\mathcal{D}}$ of $x : ?P$ in $\mathcal{D}$. Given the following two derivations:

$$\begin{array}{c}\mathcal{D}\\ \vdash \Psi, x : ?Q; \Gamma \Uparrow S\end{array} \qquad \begin{array}{c}\mathcal{E}\\ \vdash \Psi; \cdot \Uparrow Q^{\perp}\end{array}$$

we need to construct a derivation $\mathcal{F} :: \vdash \Psi; \Gamma \Uparrow S$. By applying Lemma 3.3 on $\mathcal{D}$ we can obtain a derivation $\mathcal{D}' :: \vdash \Psi, \Omega; \Delta \Downarrow Q$ and an open derivation $\mathcal{G}$ from $\vdash \Psi, \Omega; \Delta \Uparrow \cdot$ to $\vdash \Psi, x : ?Q; \Gamma \Uparrow S$. By the first induction hypothesis on $\mathcal{D}'$ and $\mathcal{E}$, we produce a derivation of $\vdash \Psi, \Omega; \Delta \Uparrow \cdot$ and compose it with $\mathcal{G}$ to obtain a derivation of $\vdash \Psi, x : ?Q; \Gamma \Uparrow S$ in which x has a lower multiplicity, and finally by applying the second induction hypothesis, we get the desired derivation. □

In order to describe the computational behaviour of the lemmas and of the procedure specified by the proof of cut elimination, within the language of $\lambda\overline{\pi}$, we introduce the following notations for *term contexts* and *value contexts* respectively:

$$\begin{array}{rcl} T & ::= & - \mid \lambda x.T \mid x\ V \mid T\ p \mid t\ V \mid \pi.T \mid \lambda !x.T \mid !x\ V \\ V & ::= & \sim \mid \lfloor T \rfloor \mid (V, q) \mid (q, V) \mid !T \end{array}$$

We use this definition to give two kinds of contexts, depending on whether the "hole" is $-$ or $\sim$. We use $T\langle t\rangle$ and $T[p]$ for the result of substituting t and p for $-$ and $\sim$ respectively, where the difference in brackets indicates which kind of hole is intended. The decomposition lemmas can now be stated in terms of the contexts:

Corollary 3.5 (Term decomposition) *Well-typed $\lambda\overline{\pi}$-terms are such that:*

(i) *a term t containing a free, linearly occurring variable x can be decomposed either into a context $T\langle-\rangle$ and a value p such that $t = T\langle x\ p\rangle$, or into a context $T[\sim]$, such that $t = T[x]$.*

(ii) *for a term t and a free exponential variable x, either x does not occur in t or t can be decomposed into a context $T\langle-\rangle$ and a value p such that $t = T\langle !x\ p\rangle$.*

When reducing the term $(\lambda x.t)\ \lfloor u \rfloor$, we first decompose t into $T\langle x\ p\rangle$, then we create the inner cut $(u\ p)$, and finally we plug it inside the context again, yielding $T\langle u\ p\rangle$ — note that this amounts to substituting u for x inside t. Written using contexts, the full set of reduction rules for $\lambda\overline{\pi}$ is thus:

$$\begin{array}{rcl} (\lambda x.T[x])\ y & \to & T[y] \\ (\lambda x.T\langle x\ p\rangle)\ \lfloor u \rfloor & \to & T\langle u\ p\rangle \\ (\pi.t)\ (p,q) & \to & (t\ p)\ q \\ (\lambda !x.T\langle !x\ p\rangle)\ !u & \to & (\lambda !x.T\langle u\ p\rangle)\ !u \\ (\lambda !x.t)\ !u & \to & t \quad \textit{if}\ x \notin \mathrm{fv}(t) \end{array}$$

where $\mathrm{fv}(t)$ denotes, as usual, the set of all *free variables* of t. The rewrite system obtained bears a striking resemblance to the linear substitution calculus defined by Accattoli *et al.* [2], which is itself based on ideas from linear logic and proof-nets. We leave the investigation of this connection as future work: for the purposes of this presentation, it suffices to note that the above reductions for λ and $\lambda!$ implement the usual notion of capture-avoiding substitution. We can therefore summarise reduction in $\lambda\overline{\pi}$ using implicit substitution, which yields a system containing the following four rules:

$$\begin{array}{rcl} (\lambda x.t)\ y & \to & t\{y/x\} \\ (\lambda x.t)\ \lfloor u \rfloor & \to & t\{u/x\} \\ (\pi.t)\ (p,q) & \to & (t\ p)\ q \\ (\lambda !x.t)\ !u & \to & t\{u/x\} \end{array} \tag{2}$$

and we observe that the first two correspond to the basic polarised subsystem of **MELL**, while the two others match the multiplicative and exponential subsystems respectively. Note that we consider here the typed fragment of the language — in the untyped case, more terms would be accepted and a thunk $\lfloor u \rfloor$ could be plugged inside a value.

Beyond cut elimination, a focused and explicitly polarised system such as ours can be given another form of dynamics, which does not represent computation as β-reduction does, but rather corresponds to a form of simplification of terms. Indeed, explicit polarity shifts can be used to introduce *delays*, compounds formed by a pair of opposite shifts, that can break a focusing phase or prevent maximal inversion. This can be interpreted as placing a piece of computation across a value, splitting this value in two: removing a delay therefore contributes to the production of a simpler, more compact term, with a different computational behaviour.

From the opposite viewpoint, it is clear that any *unfocused* proof can be mapped to a *delayed*, focused proof. With this, we can now restate the focalisation result in terms of the admissibility of *delay elimination* [8,24], rather than state it in terms of completeness with respect to an unfocused reference system. In stating the result, we use the notion of *formula context*, written $\xi\{-\}$. The formula $\xi\{P\}$ should be interpreted as a formula (positive or negative) with a linear occurrence of the subformula P. The outer context $\xi\{-\}$ then represents the *path* through $\xi\{P\}$ on which this subformula appears.

Lemma 3.6 (Positive delay elimination) *For any $\mathcal{D} :: \vdash \Psi; \Gamma, x : \uparrow\xi\{\downarrow\!\uparrow P\} \Uparrow S$ cut-free, there exists a derivation $\mathcal{E} :: \vdash \Psi; \Gamma, x : \uparrow\xi\{P\} \Uparrow S$.*

Proof. Before we prove this statement, we will make some simplifying assumptions. First, note that the nature of the formula in the context $\xi\{-\}$ only becomes relevant when the formula becomes focused. We will therefore assume that $\xi\{\downarrow\!\uparrow P\}$ contains only positive formulas between the topmost connective and the location of the subformula $\downarrow\!\uparrow P$. When $\xi\{\downarrow\!\uparrow P\}$ is selected as the focus — which we may assume appears as final rule in $\mathcal{D}$ — we can thus decompose the derivation $\mathcal{D}$ into a subderivation $\mathcal{D}' :: \vdash \Psi; \Gamma' \Downarrow \downarrow\!\uparrow P$ and an open derivation $\mathcal{G}$ from $\vdash \Psi; \Gamma' \Downarrow \downarrow\!\uparrow P$ to $\vdash \Psi; \Gamma, \uparrow\xi\{\downarrow\!\uparrow P\} \Uparrow \cdot$. Note that as $\mathcal{G}$ can only split the context, we must have $\Gamma = \Gamma', \Delta$ for some context Δ. Furthermore, composing any proof $\vdash \Psi; \Gamma'' \Downarrow Q$ with this open derivation will yield a proof of $\vdash \Psi; \Gamma'', \Delta \Downarrow \xi\{Q\}$.

By appealing to inversion twice on $\mathcal{D}'$, we get a derivation $\mathcal{D}'' :: \vdash \Psi; \Gamma', \uparrow P \Uparrow \cdot$ and by applying the linear decomposition lemma, this can be decomposed into a derivation $\mathcal{F} :: \vdash \Psi, \Psi'; \Gamma'' \Downarrow P$ and an open derivation $\mathcal{G}'$ from $\vdash \Psi, \Psi'; \Gamma'' \Uparrow \cdot$ to $\vdash \Psi, \Psi'; \Gamma' \Uparrow \cdot$ so that can now string together these derivations. From $\mathcal{F}$ and $\mathcal{G}$, we get a derivation of $\vdash \Psi, \Psi'; \Gamma'', \Delta \Downarrow \xi\{P\}$, and hence a derivation of $\vdash \Psi, \Psi'; \Gamma'', \Delta, \uparrow\xi\{P\} \Uparrow \cdot$ that we compose with $\mathcal{G}'$ to get $\vdash \Psi; \Gamma, \uparrow\xi\{P\} \Uparrow \cdot$, where we use the equality $\Gamma = \Gamma', \Delta$. □

Lemma 3.7 (Negative delay elimination) *For any $\mathcal{D} :: \vdash \Psi; \Gamma \Uparrow S_1, \xi\{\uparrow\!\downarrow N\}, S_2$ cut-free, there exists a derivation $\mathcal{E} :: \vdash \Psi; \Gamma \Uparrow S_1, \xi\{N\}, S_2$.*

Proof. The proof proceeds in a similar manner, and we only sketch the argument. First, we may assume that $\xi\{\uparrow\!\downarrow N\}$ is the first formula in the inversion context, and that only negative formulas occur on the path to $\uparrow\!\downarrow N$. Then we do a decomposition until $\uparrow\!\downarrow N$ is placed in the unordered context, and find the subderivation where it is focused again. This subderivation must immediately decompose N, and hence we can transport this decomposition down, and compose it with the context $\xi\{-\}$ to get the desired proof. □

The focalisation result is obtained by iterated application of the previous lemmas: in particular, removing negative delays forces $\parr$ connectives to be decomposed eagerly, and removing positive delays groups instances of the $\otimes$ rule. On a computational level, the procedures specified by the proofs of these two lemmas correspond to some reorganisation of a $\lambda\overline{\pi}$-term that removes an unnecessary intermediate name.

In order to precisely described the effect of delay elimination on terms, we need to refine our language of contexts to consider *monopolar* contexts, that can only contain one *pole*, which is a layer of contiguous negative or positive connectives:

$$\begin{aligned} A &::= \ - \ \mid \ (A, p) \ \mid \ (p, A) \\ B &::= \ - \ \mid \ \lambda x.B \ \mid \ \lambda !x.B \ \mid \ \pi.B \end{aligned}$$

Using these notions of contexts, we can precisely describe the effect of positive and negative delay elimination respectively:

$$\begin{array}{rcl} y\ A\langle \lfloor \lambda x.T\langle x\ p\rangle \rfloor \rangle & \to & T\langle y\ A\langle p\rangle\rangle \\ \lambda x.T\langle x\ \lfloor B\langle y\ p\rangle \rfloor \rangle & \to & B\langle T\langle y\ p\rangle\rangle \end{array}$$

embodied by rewrite rules that can apply anywhere inside a T context. Observe that in the basic case, where A is empty, the first rule is just $y\ \lfloor \lambda x.T\langle x\ p\rangle \rfloor \to T\langle y\ p\rangle$, which corresponds to the start of a focus phase immediately aborted.

4 Fragments of Linear Logic and their λ-calculi

We have seen in the previous section how the focused sequent calculus for polarised **MELL** corresponds to a refined linear λ-calculus. However, it may be easier to grasp the expressivity of the $\lambda\overline{\pi}$-calculus by considering subcalculi generated by specific subsets of valid formulas and sequents, or restrictions on rules.

Purely linear fragment. The simplest fragment that can be considered is obtained by ignoring the exponentials and the persistent context. This yields a subset of $\lambda\overline{\pi}$ that can be related to the purely linear λ-calculus — that is, the fragment of the λ-calculus where variables must be used exactly once. More precisely, there is a relatively simple encoding of the linear version of the $\overline{\lambda}$-calculus proposed by Herbelin as an interpretation of the **LJT** focused intuitionistic sequent calculus [12]. This calculus has lists of arguments in applications, and is based on the following syntax:

$$t, u \ ::= \ x\ k \mid \lambda x.t \mid t\ k \qquad\qquad k, m \ ::= \ \varepsilon \mid t :: k$$

where ε represents an empty list of arguments, needed to recover the simple variable x, encoded here as $x\ \varepsilon$, and $::$ is the list constructor. The type system for this calculus, in its linear form, is given by the linear variant of **LJT** that we call **IMLLT**, where two kinds of sequents are distinguished: we write $\Gamma \vdash N$ for an unfocused sequent and $\Gamma, [N] \vDash a$ for a sequent focusing on the left on N, where the right-hand side is limited to an atom because we consider η-long terms. The encoding is based on a simple translation of unpolarised intuitionistic linear formulas, that we denote by A or B:

$$[\![a]\!] \ = \ \uparrow a \qquad\qquad [\![A \multimap B]\!] \ = \ \uparrow [\![A]\!]^{\perp} \parr [\![B]\!]$$

which can be extended pointwise to contexts. The most important part of the translation relates sequents of **IMLLT** to sequents of our system system where the persistent context is always empty and thus omitted:

$$\Gamma \vdash A \quad \rightsquigarrow_{\boldsymbol{JT}} \quad \vdash \uparrow[\![\Gamma]\!]^{\perp} \Uparrow [\![A]\!] \qquad\qquad \Gamma, [B] \vDash a \quad \rightsquigarrow_{\boldsymbol{JT}} \quad \vdash \uparrow[\![\Gamma]\!]^{\perp}, c : \uparrow a \Downarrow [\![B]\!]^{\perp}$$

where by convention we choose c as the name of the atomic right-hand side of any focused **IMLLT** sequent — this variable will represent the empty list. The translation of the typing derivations is then based on these translations, as follows:

$$\frac{}{[a] \vDash \varepsilon : a} \quad \rightsquigarrow_{\boldsymbol{JT}} \quad \frac{}{c \vdash c : \uparrow a \Downarrow \downarrow \overline{a}}$$

$$\frac{\Gamma, [B] \vDash k : a}{\Gamma, x : B \vdash x\ k : a} \quad \leadsto_{\boldsymbol{JT}} \quad \cfrac{\cfrac{p \vdash \uparrow\llbracket\Gamma\rrbracket^{\perp}, c : \uparrow a \Downarrow \llbracket B\rrbracket^{\perp}}{x\ p \vdash \uparrow\llbracket\Gamma\rrbracket^{\perp}, x : \uparrow\llbracket B\rrbracket^{\perp}, c : \uparrow a \Uparrow \cdot}}{\lambda c.(x\ p) \vdash \uparrow\llbracket\Gamma\rrbracket^{\perp}, x : \uparrow\llbracket B\rrbracket^{\perp} \Uparrow \uparrow a}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \multimap B} \quad \leadsto_{\boldsymbol{JT}} \quad \cfrac{\cfrac{u \vdash \uparrow\llbracket\Gamma\rrbracket^{\perp}, x : \uparrow\llbracket A\rrbracket^{\perp} \Uparrow \llbracket B\rrbracket}{\lambda x.u \vdash \uparrow\llbracket\Gamma\rrbracket^{\perp} \Uparrow \uparrow\llbracket A\rrbracket^{\perp}, \llbracket B\rrbracket}}{\pi.\lambda x.u \vdash \uparrow\llbracket\Gamma\rrbracket^{\perp} \Uparrow \uparrow\llbracket A\rrbracket^{\perp} \parr \llbracket B\rrbracket}$$

$$\frac{\Gamma \vdash t : B \quad \Delta, [C] \vDash k : a}{\Gamma, \Delta, [B \multimap C] \vDash t :: k : a} \quad \leadsto_{\boldsymbol{JT}} \quad \cfrac{\cfrac{u \vdash \uparrow\llbracket\Gamma\rrbracket^{\perp} \Uparrow \llbracket B\rrbracket}{\lfloor u\rfloor \vdash \uparrow\llbracket\Gamma\rrbracket^{\perp} \Downarrow \downarrow\llbracket B\rrbracket} \quad p \vdash \uparrow\llbracket\Delta\rrbracket^{\perp}, c : \uparrow a \Downarrow \llbracket C\rrbracket^{\perp}}{(\lfloor u\rfloor, p) \vdash \uparrow\llbracket\Gamma\rrbracket^{\perp}, \uparrow\llbracket\Delta\rrbracket^{\perp}, c : \uparrow a \Downarrow \downarrow\llbracket B\rrbracket \otimes \llbracket C\rrbracket^{\perp}}$$

where p is the translation of k, and u is the translation of t. The syntax $\uparrow\Gamma$ denotes a context Γ where formulas are annotated with a negative shift. This translation turns lists of arguments into right-associated pairs of thunks, and translating abstractions requires the π operator. There is therefore a matching between these constructs, and we can compose such terms in the expected way with the cut rule, corresponding to the main head cut of **LJT** in its linear form:

$$\frac{\Gamma \vdash t : N \quad \Delta, [N] \vDash k : a}{\Gamma, \Delta \vdash t\ k : a} \quad \leadsto_{\boldsymbol{JT}} \quad \cfrac{\cfrac{u \vdash \uparrow\llbracket\Gamma\rrbracket^{\perp} \Uparrow \llbracket N\rrbracket \quad p \vdash \uparrow\llbracket\Delta\rrbracket^{\perp}, c : \uparrow a \Downarrow \llbracket N\rrbracket^{\perp}}{u\ p \vdash \uparrow\llbracket\Gamma\rrbracket^{\perp}, \uparrow\llbracket\Delta\rrbracket^{\perp}, c : \uparrow a \Uparrow \cdot}}{\lambda c.(u\ p) \vdash \uparrow\llbracket\Gamma\rrbracket^{\perp}, \uparrow\llbracket\Delta\rrbracket^{\perp} \Uparrow \uparrow a}$$

so that reduction in this linear $\overline{\lambda}$-calculus is simulated by reduction in our focused **MELL** system. More specifically, the reduction system $\rightarrow_{\boldsymbol{JT}}$ that we use for **IMLLT** is based on implicit substitution rather than the original explicit ones in **LJT** [12]:

$$\begin{array}{rcl} (\lambda x.t)\ (u :: k) & \rightarrow_{\boldsymbol{JT}} & t\{u/x\}\ k \\ (\lambda x.t)\ \varepsilon & \rightarrow_{\boldsymbol{JT}} & t \\ (t\ k)\ m & \rightarrow_{\boldsymbol{JT}} & t\ (k \mathbin{@} m) \end{array} \tag{3}$$

where @ denotes the concatenation of lists. In the $\lambda\overline{\pi}$-calculus, we consider reduction rules in their compact form, as shown in (2). As a result, we obtain the simple simulation of this linear λ-calculus in $\lambda\overline{\pi}$ described by the theorem below.

Theorem 4.1 *If $t \leadsto_{\boldsymbol{JT}} u$ and $t \rightarrow_{\boldsymbol{JT}} v$ there is a w such that $v \leadsto_{\boldsymbol{JT}} w$ and $u \rightarrow^{*} w$.*

Proof. We proceed by structural induction on the term t, extending the statement to handle the translation of lists as $\lambda\overline{\pi}$-terms. In the base case, it is the empty list and none of the reduction rules apply, so we are done. In general, all cases except for the redex $t\ k$ directly rely on the induction hypothesis. In this last case, we consider possible reductions, so that the compound reductions:

$$\begin{array}{rcl} \lambda c.((\pi.\lambda x.t)\ (\lfloor u\rfloor, q)) & \rightarrow^{*} & \lambda c.(t\{u/x\}\ q) \\ \lambda c.((\lambda x.t)\ c) & \rightarrow^{*} & \lambda c.t\{c/x\} \\ \lambda c.((\lambda d.u\ p)\ q) & \rightarrow^{*} & \lambda c.(u\ p\{q/d\}) \end{array}$$

are just the translations of the reduction rules shown in (3). The first reduction is simple and relies on the decomposition of the pair encoding the :: constructor, and a substitution. The second reduction just performs a substitution, but one should note that after reduction, c is no longer a right-hand side marker, but simply a renaming of x. Finally, the last reduction relies on the encoding of lists as right-associated pairs, so that $p\{q/d\}$ is exactly the encoding of $k \mathbin{@} m$ in $\lambda\overline{\pi}$. □

Notice that this encoding translates primitive constructs in $\overline{\lambda}$ into compound constructs of the $\lambda\overline{\pi}$-calculus. Some terms in our calculus have no equivalent in the interpretation of **IMLLT**: we have captured here only an *intuitionistic* fragment of **MLL**. This can be seen quite clearly in our translation, as it corresponds to the presence of the single variable c with a type of the shape $\uparrow a$ in the context, which represents the unique right-hand side of a sequent. We can capture a larger, more *classical* fragment of the calculus, but makes the embedding slightly more complex. Although we could study encodings of a linear variant of the $\lambda\mu$-calculus, we choose to use exponentials to represent the full sequent-based variant of $\lambda\mu$.

Exponential fragment. Beyond the purely linear fragment discussed above, an obvious subsystem of interest is the one where no polarity shifts appear, related to **LLP** [15], and one might want to push this further and try to eliminate the need for a linear context altogether. This is problematic since the axiom rule applies only when the linear context is not empty, but it is possible to work around this problem by considering the exponential axiom rule from (1) obtained from the identity expansion result:

$$\frac{}{\;!x \vdash \Psi, x : ?P \Downarrow !P^{\perp}\;}$$

However, it is not possible to avoid entirely polarity shifts in our system, since atoms are not handled without them. We consider the $\overline{\lambda}\mu$-calculus of Herbelin [13], which is an interpretation of **LKT**, and can be obtained by adding the control rules from $\lambda\mu$ into the **LJT** system and term assignment. Due to the polarised setting in **MELL**, we use an explicitly polarised version of this calculus, where the μ and naming rules are reflected on types by shifts — that we write $\downharpoonleft$ and $\upharpoonleft$ to distinguish them from the shifts of **MELL**. The translation of formulas is then defined as:

$$[\![a]\!] \;=\; ?{\downarrow}\overline{a} \qquad [\![P \supset N]\!] \;=\; ?[\![P]\!]^{\perp} \parr [\![N]\!] \qquad [\![\downharpoonleft N]\!] \;=\; ![\![N]\!] \qquad [\![\upharpoonleft P]\!] \;=\; ?[\![P]\!]$$

The **LKT** system has two kinds of sequents, just as **LJT**, which are obtained by adding the context Δ of other right-hand sides. Since we used polarised formulas, the context Γ in the left-hand side always contains formulas of the shape $\downharpoonleft N$, just as Δ. Sequents are then encoded using the translation of formulas as follows:

$$\begin{array}{rcl} \Gamma \vdash N \mid \Delta & \leadsto_{\boldsymbol{KT}} & \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!] \Uparrow [\![N]\!] \\ \Gamma, [N] \vDash a \mid \Delta & \leadsto_{\boldsymbol{KT}} & \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!], c : ?{\downarrow}\overline{a} \Downarrow [\![N]\!]^{\perp} \end{array}$$

where focused **MELL** sequents are written without a linear context, since it will always be empty in this translation. Indeed, polarity shifts, needed to introduce formulas in the linear context, are used only on atoms and therefore they cannot be treated out of the axiom rule. The control rules of the calculus, concerning μ and naming, are translated as follows in our system:

$$\frac{\Gamma \vdash t \mid \alpha : \downharpoonleft N, \Delta}{\Gamma \vdash \mu\alpha.t : \upharpoonleft\downharpoonleft N \mid \Delta} \quad \rightsquigarrow_{\boldsymbol{KT}} \quad \frac{u \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!], c : ?![\![N]\!] \Uparrow \cdot}{\lambda !c.u \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!], \Uparrow ?![\![N]\!]}$$

$$\frac{\Gamma \vdash t : N \mid \alpha : \downharpoonleft N, \Delta}{\Gamma \vdash [\alpha]\ t \mid \alpha : \downharpoonleft N, \Delta} \quad \rightsquigarrow_{\boldsymbol{KT}} \quad \cfrac{\cfrac{u \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!], c : ?![\![N]\!] \Uparrow [\![N]\!]}{!u \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!], c : ?![\![N]\!] \Downarrow ![\![N]\!]}}{!c\ !u \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!], c : ?![\![N]\!] \Uparrow \cdot}$$

where c is the name given to the marker labelled α in **LKT**, and u is the translation of t. The other rules are translated in a way very similar to the encoding for **LJT**, but they all have the generalised treatment of the right-hand side context Δ:

$$\frac{}{\Gamma, [a] \vDash \varepsilon : a \mid \Delta} \quad \rightsquigarrow_{\boldsymbol{KT}} \quad \frac{}{!c \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!], c : ?\downarrow\bar{a} \Downarrow !\uparrow a}$$

$$\frac{\Gamma, [N] \vDash k : a \mid \Delta}{\Gamma, x : \downharpoonleft N \vdash x\ k : a \mid \Delta} \quad \rightsquigarrow_{\boldsymbol{KT}} \quad \cfrac{\cfrac{p \vdash [\![\Gamma]\!]^{\perp}, x : ?[\![N]\!]^{\perp}, ?[\![\Delta]\!], c : ?\downarrow\bar{a} \Downarrow [\![N]\!]^{\perp}}{!x\ p \vdash [\![\Gamma]\!]^{\perp}, x : ?[\![N]\!]^{\perp}, ?[\![\Delta]\!], c : ?\downarrow\bar{a} \Uparrow \cdot}}{\lambda !c.(!x\ p) \vdash [\![\Gamma]\!]^{\perp}, x : ?[\![N]\!]^{\perp}, ?[\![\Delta]\!] \Uparrow ?\downarrow\bar{a}}$$

$$\frac{\Gamma, x : \downharpoonleft N \vdash t : M \mid \Delta}{\Gamma \vdash \lambda x.t : \downharpoonleft N \supset M \mid \Delta} \quad \rightsquigarrow_{\boldsymbol{KT}} \quad \cfrac{\cfrac{u \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!], x : ?[\![N]\!]^{\perp} \Uparrow [\![M]\!]}{\lambda !x.u \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!] \Uparrow ?[\![N]\!]^{\perp}, [\![M]\!]}}{\pi.\lambda !x.u \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!] \Uparrow ?[\![N]\!]^{\perp} \parr [\![M]\!]}$$

$$\frac{\Gamma \vdash t : N \mid \Delta \qquad \Gamma, [M] \vDash k : a \mid \Delta}{\Gamma, [\downharpoonleft N \supset M] \vDash t :: k : a \mid \Delta} \quad \rightsquigarrow_{\boldsymbol{KT}}$$

$$\cfrac{\cfrac{u \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!], c : ?\downarrow\bar{a} \Uparrow [\![N]\!]}{!u \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!], c : ?\downarrow\bar{a} \Downarrow ![\![N]\!]} \qquad p \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!], c : ?\downarrow\bar{a} \Downarrow [\![M]\!]^{\perp}}{(!u, p) \vdash [\![\Gamma]\!]^{\perp}, ?[\![\Delta]\!], c : ?\downarrow\bar{a} \Downarrow ![\![N]\!] \otimes [\![M]\!]^{\perp}}$$

Observe that the left implication rule is compound in our presentation of this calculus, but so is the translation, in the same way. The key idea here is that more than a single right-hand side marker can be used in a single sequent, due to the classical setting. But in the $\lambda\overline{\pi}$-calculus itself, control does not need to use these tools: it is the consequence of the shape of typing rules, and in particular of the continuation behaviour of the focus rule. Indeed, if a variable of type $\uparrow P$ is available, it can be applied to a value typed by P itself in a focused phase. The continuation behaviour of variable application is well illustrated by the a special case of positive delay elimination:

$$y\ \lfloor \lambda x.T\langle x\ p\rangle \rfloor \quad \to \quad T\langle y\ p\rangle$$

which can be read as $y\ \lfloor \lambda x.t \rfloor \ \to t\{y/x\}$, so that we see t using the name of the variable to which it was given as argument.

Finally, the question of simulating the reduction of the $\overline{\lambda}\mu$-calculus using this encoding is more complex than in the purely linear case. Indeed, the reduction rule for μ is performing a relatively complex operation:

$$(\mu\alpha.t)\ k \quad \to_{\boldsymbol{KT}} \quad \mu\alpha.t\{[\alpha](u\ k)/[\alpha]\ u\}$$

that can be observed in $\lambda\overline{\pi}$ if the proper cut and shift rules are introduced, but these yield problems concerning the preservation of types of different subterms during the reduction process.

5 Conclusion and Future Work

We presented a Curry-Howard interpretation of focused **MELL**, with a novel proof of cut elimination based on reducing cuts at a distance. The variant of the λ-calculus obtained is more similar to the usual λ-calculus than other calculi based on sequent calculi and it offers a simple syntax for **MELL** proofs. Moreover, this system has connections to some well-known variations of the λ-calculus, and its reduction simulates in one step the usual notion of β-reduction. The investigation of the computational meaning of a focused cut elimination is important, as it relates to the question of evaluation strategies and has a nice proof-theoretic behaviour.

As mentioned, the system of reductions we present bears a striking resemblance to that of the *linear substitution calculus*, which employs a similar notion of reduction *at a distance*. It would be interesting to observe how deep this similarity is, and whether the **LSC** can be generalised based on the proof-theoretic approach presented here. In a similar vein, it would be interesting to have a computational interpretation of focused proofs in the presence of the remaining connectives of linear logic, in particular the additives, which are known to introduce a notion of case analysis. Richer logics could also be considered, such as μ**MALL** [5], where induction and coinduction are supported directly in an elegant, proof-theoretic way. The use of fixpoints as alternative to persistent variables yields many questions, concerning for example the use of this system as a programming language. In terms of language it would also, of course, be interesting — and surely straightforward — to extend this interpretation to second-order quantifiers, reaching the expressivity of System **F**.

Finally, it is well-known that using delays, many other calculi can be represented as appropriately polarised fragments of a strongly focused sequent calculus. This is seen for example in [18] where **LJT** and **LJQ** are both shown to be representable as fragments of **LJF**. Given the generality of our calculus, it is possible that it could serve as a *lingua franca* for the large variety of classical λ-calculi found in the literature. Exploring such a possibility will pinpoint the general position of linear logic in the field, and highlight how it provides a crucial tool in the understanding of computational phenomena.

Acknowledgement

The authors would like to thank the anonymous reviewers for their helpful and thorough comments on a previous draft.

References

[1] Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.

[2] Beniamino Accattoli, Eduardo Bonelli, Delia Kesner, and Carlos Lombardi. A nonstandard standardization theorem. In S. Jagannathan and P. Sewell, editors, *POPL'14*, pages 659–670, 2014.

[3] Beniamino Accattoli and Ugo Dal Lago. Beta reduction is invariant, indeed. In T. Henzinger and D. Miller, editors, *CSL-LICS'14*, 2014.

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

[5] Dabid Baelde and Dale Miller. Least and greatest fixed points in linear logic. In N. Dershowitz and A. Voronkov, editors, *LPAR'07*, volume 4790 of *LNCS*, pages 92–106, 2007.

[6] Gianluigi Bellin and Philip Scott. On the pi-calculus and linear logic. *Theoretical Computer Science*, 135:11–65, 1994.

[7] Gavin Bierman. *On Intuitionistic Linear Logic.* PhD thesis, University of Cambridge, 1993.

[8] Kaustuv Chaudhuri, Nicolas Guenot, and Lutz Straßburger. The focused calculus of structures. In M. Bezem, editor, *CSL'11*, volume 12 of *LIPIcs*, pages 159–173, 2011.

[9] Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In *ICFP'00: Proceedings of the fifth ACM SIGPLAN international conference on Functional programming*, pages 233–243, 2000.

[10] Roy Dyckhoff and Stéphane Lengrand. LJQ: a strongly focused calculus for intuitionistic logic. In A. Beckmann, U. Berger, B. Löwe, and J. Tucker, editors, *CiE'06*, volume 3988 of *LNCS*, pages 173–185, 2006.

[11] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[12] Hugo Herbelin. A λ-calculus structure isomorphic to Gentzen-style sequent calculus structure. In L. Pacholski and J. Tiuryn, editors, *CSL'94*, volume 933 of *LNCS*, pages 61–75, 1994.

[13] Hugo Herbelin. *Séquents qu'on calcule: de l'interprétation du calcul des séquents comme calcul de lambda-termes et comme calcul de stratégies gagnantes.* PhD thesis, Université Paris 7, 1995.

[14] Delia Kesner and Stéphane Lengrand. Resource operators for the λ-calculus. *Information and Computation*, 205(4):419–473, 2007.

[15] Olivier Laurent. *Étude de la polarisation en logique.* Thèse de doctorat, Université Aix-Marseille II, 2002.

[16] Olivier Laurent. Polarized proof-nets and $\lambda\mu$-calculus. *Theoretical Computer Science*, 290(1):161–188, 2003.

[17] Paul Blain Levy. Call-by-push-value: Decomposing call-by-value and call-by-name. *Higher-Order and Symbolic Computation*, 19(4):377–414, 2006.

[18] Chuck Liang and Dale Miller. Focusing and polarization in intuitionistic logic. In J. Duparc and T. A. Henzinger, editors, *CSL'07*, volume 4646 of *LNCS*, pages 451–465. Springer, 2007.

[19] Guillaume Munch-Maccagnoni. Focalisation and classical realisability. In E. Grädel and R. Kahle, editors, *CSL'2009*, volume 5771 of *LNCS*, pages 409–423, 2009.

[20] Michel Parigot. $\lambda\mu$-calculus: An algorithmic interpretation of classical natural deduction. In *LPAR'92*, volume 624 of *LNCS*, pages 190–201, 1992.

[21] Robert J. Simmons. Structural focalization. *ACM Trans. Comput. Log.*, 15(3):21:1–21:33, 2014.

[22] Philip Wadler. A syntax for linear logic. In S. Brookes, M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *MFPS'93*, volume 802 of *LNCS*, pages 513–529, 1993.

[23] Philip Wadler. Propositions as sessions. *Journal of Functional Programming*, 24(2-3):384–418, 2014.

[24] Stéphane Zimmermann. *Vers une Ludique Différentielle.* Thèse de doctorat, Université Paris Diderot, 2013.