

Least and greatest fixed points in linear logic

Extended Version

September 13, 2007

David Baelde and Dale Miller

INRIA & LIX/École Polytechnique, Palaiseau, France

david.baelde at ens-lyon.org dale.miller at inria.fr

Abstract. The first-order theory of MALL (multiplicative, additive linear logic) over only equalities is an interesting but weak logic since it cannot capture unbounded (infinite) behavior. Instead of accounting for unbounded behavior via the addition of the exponentials (! and ?), we add least and greatest fixed point operators. The resulting logic, which we call μMALL^\pm , satisfies two fundamental proof theoretic properties. In particular, μMALL^\pm satisfies cut-elimination, which implies consistency, and has a complete focused proof system. This second result about focused proofs provides a strong normal form for cut-free proof structures that can be used, for example, to help automate proof search. We then consider applying these two results about μMALL^\pm to derive a focused proof system for an intuitionistic logic extended with induction and co-induction. The traditional approach to encoding intuitionistic logic into linear logic relies heavily on using the exponentials, which unfortunately weaken the focusing discipline. We get a better focused proof system by observing that certain fixed points satisfy the structural rules of weakening and contraction (without using exponentials). The resulting focused proof system for intuitionistic logic is closely related to the one implemented in Bedwyr, a recent model checker based on logic programming. We discuss how our proof theory might be used to build a computational system that can partially automate induction and co-induction.

1 Introduction

In order to justify the design and implementation architecture of a computational logic system, foundational results concerning the normal forms of proofs are often used. One starts with the *cut-elimination theorem* since it usually guarantees other properties of the logic (e.g., consistency) and that there is no need to automate the creation of *lemmas* during proof search. In many situations, the cut-elimination theorem implies that all formulas considered during the search for a proof are subformulas of the original, proposed theorem. This does not hold, in particular, when higher-order (relation) variables are used, which is the case in this paper where the rules for induction and co-induction use such higher-order variables. A second normal form theorem, usually related to *focused proofs* [And92] is also important to establish. Such “focusing” theorems provide normal forms that organize invertible and non-invertible inference rules into collections: such striping of the inference rules in a cut-free derivation can be used to understand which choices in building proofs might need to be reconsidered (via backtracking) and

which do not. As we shall see, focusing yields useful structure in cut-free proofs, even when the subformula property does not hold.

Various computational systems have employed different focusing theorems: much of Prolog's design and implementations can be justified by the completeness of SLD-resolution [AvE82]; uniform proofs (goal-directed proofs) in intuitionistic and intuitionistic linear logics have been used to justify λ Prolog [MNPS91] and Lolli [HM94]; the classical linear logic programming languages LO [AP91] and Forum [Mil96] have used directly Andreoli's general focusing result [And92] for linear logic.

In this paper, we establish these two foundational proof-theoretic properties for the following logic. We first extend the multiplicative and additive fragment of linear logic (MALL) with equality and quantification (via \forall and \exists) over simply typed λ -terms. Because of the bounded use of formulas during proof construction, provability in this logic, call it $MALL^\equiv$, can be reduced to deciding unification problems (under a mixed prefix) which is decidable for the first-order fragment of $MALL^\equiv$. An elegant and well known way to make this logic more expressive is to add the exponentials $!$ and $?$ and the rules of inference that allow for certain occurrences of formulas marked with these systems to be contracted and weakened [Gir87]. Such modal-like operators are not, however, without their problems. In particular, the exponentials are not canonical since there are different ways to formulate the rules for the promotion and structural rules for exponentials and some of these choices lead to different versions of logic (for example, elementary and light linear logics [Gir98] and soft linear logic [Laf04]). Even if we fix the inference rules for the exponentials, as in standard linear logic, the rules do not describe unique exponentials. If one gives a red tensor and a blue tensor the same inference rules, then one can prove that these two tensors are, in fact, equivalent. All of linear logic connectives except the exponentials yield similar theorems. It is certainly possible to consider a (partially ordered) collection of exponentials on top of MALL (see, for example, [DJS93]).

An alternative to strengthen MALL with exponentials is to extend it with fixed points. Early approaches to adding fixed points [Gir92,SH93] involved inference rules that could only unfold fixed point descriptions: as a consequence, such logics could not discriminate between a least and greatest fixed point. Stronger systems that allow induction [MM00] as well as co-induction [Tiu04,MT03] include inference rules using a higher-order variable that ranges over prefixed or postfix points (invariants). Of course, approaches that use (co)induction are not without problems as well: various restrictions on fixed point expressions and on invariants may need to be considered. In any case, we shall explore this alternative to exponentials: in particular, we extend the logic $MALL^\equiv$ to $\mu MALL^\equiv$ by adding the two fixed points μ and ν .

Besides considering fixed points as alternatives to the exponentials, there are other reasons for examining $\mu MALL^\equiv$. First, least and greatest fixed points are de Morgan duals of one another and, hence, the classical nature of linear logic should offer some economy and elegance in developing their proof theory, in contrast to intuitionistic logic. Second, since linear logic can be seen as the logic behind intuitionistic logic, it will be rather easy to develop a focusing proof system for intuitionistic logic and fixed points based on the structure of the one we develop for $\mu MALL^\equiv$.

It is important to stress that we are using linear logic here as “the logic behind computational logic” and not, as it is more traditionally understood, as the logic of resource management (in the sense of multiset rewriting, database updates, Petri nets, etc). Instead, we find the proof theory of linear logic an appropriate and powerful setting for exploring the structure of proofs in various intuitionistic logics (see [LM07] for another such use of linear logic).

In the next section, we define μMALL^\equiv and prove some of the most basic aspects of its proof theory, including the cut-elimination theorem. Section 3 presents a focused proof system that is complete for μMALL^\equiv . In Section 4 we describe a few examples of (focused) derivations in μMALL^\equiv . Section 5 shows how the proof theory of μMALL^\equiv can be applied to an intuitionistic logic extended with induction and co-induction, and to the intuitionistic logic of fixed point unfoldings that is the foundation of the recent computational system Bedwyr [BGM⁺07].

2 Linear logic extended with fixed points

For clarity, we will use simply typed λ -calculus as our language of formulas. We assume that formulas are always in $\beta\eta$ -long form. We make few restrictions on the language of terms in this work and choose simply typed λ -calculus for them as well: we assume that the reader understands the basics involving substitution, equality, and complete set of unifiers for such terms. In most of our examples variables will be of ground type, and thus the possibly infinite complete set of unifiers can be replaced by the most general unifier when there is one. Depending on one’s interests, it is possible to choose weaker (*e.g.*, first-order) or more powerful (*e.g.*, dependently typed) terms.

In the following, terms are denoted by s, t ; vectors of terms are denoted by s, \mathbf{t} ; formulas (objects of type o) are denoted by P, Q ; eigenvariables are denoted by x, c . Finally, the syntactic variable B represents a formula abstracted over by a predicate and n terms $(\lambda p \lambda x_1 \dots \lambda x_n. P p x_1 \dots x_n)$. We have the following formula constructors:

$$P ::= P \otimes P \mid P \oplus P \mid P \wp P \mid P \& P \mid \mathbf{1} \mid \mathbf{0} \mid \perp \mid \top \\ \mid \exists_\gamma x. Px \mid \forall_\gamma x. Px \mid s \stackrel{\gamma}{=} t \mid s \stackrel{\gamma}{\neq} t \mid \mu_{\gamma_1 \dots \gamma_n} B \mathbf{t} \mid \nu_{\gamma_1 \dots \gamma_n} B \mathbf{t}$$

The syntactic variable γ ranges over all simple types that do not contain o . The quantifiers have type $(\gamma \rightarrow o) \rightarrow o$ and the equality and inequality have type $\gamma \rightarrow \gamma \rightarrow o$. The connectives μ and ν have type $(\tau \rightarrow \tau) \rightarrow \tau$ where τ is $\gamma_1 \rightarrow \dots \rightarrow \gamma_n \rightarrow o$ for some arity $n \geq 0$. We shall almost always elide the references to γ , assuming that they can be determined from context when it is important to know their value. Formulas with top-level connective μ or ν are called fixed point expressions and can be arbitrarily nested. The first argument of a fixed point expression, denoted by B , is called its *body*.

Quantifiers and (in)equality are not new and play a small role in the proof theory results: they are, however, crucial for our example applications. The central feature here is the fixed point constructs. Finally, note that there are no atoms in the μMALL^\equiv grammar. We shall see in the following the advantages of using fixed points instead.

MALL rules	First-order structure
$\frac{}{\vdash \mathbf{1}} \quad \frac{\vdash \Gamma, P \quad \vdash \Delta, Q}{\vdash \Gamma, \Delta, P \otimes Q} \quad \frac{\vdash \Gamma, P, Q}{\vdash \Gamma, P \wp Q} \quad \frac{\vdash \Gamma}{\vdash \Gamma, \perp}$	$\frac{\vdash \Gamma, Pt}{\vdash \Gamma, \exists x.Px} \quad \frac{\vdash \Gamma, Pc}{\vdash \Gamma, \forall x.Px} \quad c \text{ new}$
$\frac{}{\vdash \Delta, \top} \quad \frac{\vdash \Gamma, P \quad \vdash \Gamma, Q}{\vdash \Gamma, P \& Q} \quad \frac{\vdash \Gamma, P_i}{\vdash \Gamma, P_0 \oplus P_1}$	$\frac{}{\vdash t = t} \quad \frac{\{\vdash \Gamma \theta : \theta \in csu(s \doteq t)\}}{\vdash \Gamma, s \neq t}$
Fixed points (where S is closed, \mathbf{x} is new)	
$\frac{\vdash \Gamma, B(\mu B)t}{\vdash \Gamma, \mu Bt} \mu$	$\frac{\vdash \Gamma, St \quad \vdash BS\mathbf{x}, (S\mathbf{x})^\perp}{\vdash \Gamma, \nu Bt} \nu \quad \frac{}{\vdash \mu Bt, \nu Bt} \mu\nu$

Fig. 1. Inference rules for μMALL^\equiv

Definition 1. We define the negation \overline{B} of a body B , and extend the usual definition of the involutive negation as follows:

$$\overline{B} \stackrel{def}{=} \lambda p.\lambda x.(B(\lambda x.(px)^\perp)x)^\perp \quad (s = t)^\perp \stackrel{def}{=} s \neq t \quad (\mu Bt)^\perp \stackrel{def}{=} \nu \overline{Bt}$$

A body B is said to be monotonic when for any variables p and t , the negation normal and λ -normal form of Bpt does not contain any negated instance of p .

We shall assume that *all bodies are monotonic*. In other words, negation (\bullet^\perp for formulas and \blacklozenge for bodies) is not part of the syntax since negation normal form of formulas and bodies without atoms do not contain negations and since we forbid them explicitly in fixed point expressions. When we write negation in some inference rules, we shall be considering it as implicitly computing the negation normal form.

The monotonicity of a function is also a natural condition for the existence of fixed points in lattices or other models. The condition of monotonicity is used only syntactically here since we are not studying the semantics of μMALL^\equiv .

We present the inference rules for μMALL^\equiv in Figure 1. The initial rule is restricted to fixed points. In the ν rule, which provides both induction and coinduction, S is called the (co)invariant, and has the same type as νB , of the form $\gamma_1 \rightarrow \dots \rightarrow \gamma_n \rightarrow o$. The treatment of equality dates back to [Gir92,SH93]. In the inequality rule, csu stands for complete set of unifiers. This set has at most one element in the first-order case, but can be infinite in presence of higher-order term variables, which we do not exclude. In that case, the proofs are infinitely branching but still have a finite depth. They are handled easily in our proofs by means of transfinite inductions. Again, the use of higher-order terms, and even the presence of the equality connectives are not essential to this work. All the results presented below hold in the logic without equality, and they do not make much assumptions on the language of terms.

Proposition 1. The following inference rules are derivable:

$$\frac{}{\vdash P, P^\perp} \text{init} \quad \frac{\vdash \Gamma, B(\nu B)t}{\vdash \Gamma, \nu Bt} \nu R$$

Proof The admissibility of *init* is a standard result. It is proved by induction on F , the base case being the fixed points. The unfolding νR is derivable from ν , using the body of the definition $B(\nu B)$ as the invariant S . The proof of co-invariance $(B(B(\nu B))x, (B(\nu B)x)^\perp)$ is by induction on B . Thanks to its monotonicity, non-trivial branches end by the following derivation:

$$\frac{\frac{\overline{\vdash B\nu Bx, \overline{B\mu Bx}}}{\vdash B\nu Bx, \mu Bx} \textit{init}}{\vdash B\nu Bx, \mu Bx} \mu$$

□

Let's assume the admissibility of the *cut* rule for a moment, and prove a few interesting result.

Proposition 2. *The deep cut under a monotonic context B is admissible (x is new):*

$$\frac{\vdash \Gamma, BQ\mathbf{t} \quad \vdash (Qx)^\perp, Px}{\vdash \Gamma, BP\mathbf{t}} \textit{deep}$$

Proof Building a derivation of $BQ\mathbf{t} \multimap BP\mathbf{t}$ from the derivation of $Qx \multimap Px$ relies on the dualities of the logic. It is proved by induction on the number of fixed point connectives surrounding p in Bpx , with a subinduction on the size of B . We only describe the step for the fixed points, which reduces the property for $\lambda p.\nu(Bp)\mathbf{t}$ to that for $\lambda p.Bp(\nu(BQ)x)$.

$$\frac{\frac{\overline{\vdash (\nu(BQ)\mathbf{t})^\perp, \nu(BQ)\mathbf{t}}}{\vdash (\nu(BQ)\mathbf{t})^\perp, \nu(BP)\mathbf{t}} \textit{init} \quad \frac{\overline{\vdash (BQ(\nu(BQ))x)^\perp, BP(\nu(BQ))x}}{\vdash (\nu(BQ)x)^\perp, BP(\nu(BQ))x} \mu}{\vdash (\nu(BQ)\mathbf{t})^\perp, \nu(BP)\mathbf{t}} \nu$$

□

Definition 2. *We classify as asynchronous (resp. synchronous) the connectives $\wp, \perp, \&, \top, \forall, \neq, \nu$ (resp. $\otimes, \mathbf{1}, \oplus, \mathbf{0}, \exists, =, \mu$). A formula is said to be asynchronous (resp. synchronous) when its top-level connective is asynchronous (resp. synchronous). A formula is said to be fully asynchronous (resp. fully synchronous) when all of its connectives are asynchronous (resp. synchronous). Finally, a body $\lambda p\lambda x.Bpx$ is said to be fully asynchronous (resp. fully synchronous) when the formula Bpx is fully asynchronous (resp. fully synchronous).*

Notice, for example, that $\lambda p\lambda x.px$ is fully asynchronous and fully synchronous. The next proposition plays a central role in the focusing proof system presented in Section 3 and is crucial for our encoding in Section 5 of intuitionistic logic extended with least and greatest fixed points.

Proposition 3. *The following structural rules are admissible provided that B is fully asynchronous:*

$$\frac{\vdash \Gamma, \nu B\mathbf{t}, \nu B\mathbf{t}}{\vdash \Gamma, \nu B\mathbf{t}} \nu C \quad \frac{\vdash \Gamma}{\vdash \Gamma, \nu B\mathbf{t}} \nu W$$

Hence, the following structural rules hold for any fully asynchronous formula P :

$$\frac{\vdash \Gamma, P, P}{\vdash \Gamma, P} C \quad \frac{\vdash \Gamma}{\vdash \Gamma, P} W$$

Proof We first prove the admissibility of νW . It is obtained by co-induction, choosing \perp as the co-invariant. We obtain the co-invariance proof from a more general result: for any family of monotonic and fully asynchronous contexts $(B_i)_i$, it is provable that $((B_i \perp t)_i, \mathbf{1})$. This is done by induction on the total size of the family. The proof is trivial if the family is empty. If $B_i \perp t$ is an inequality we conclude by induction with a new t' ; if it's \top our proof is done; if it's \perp (being a recursive occurrence or not) this B_i disappears. The \wp case is done by induction hypothesis, the resulting family has more bodies but is smaller; the $\&$ makes use of two instances of the induction hypothesis. Finally, the ν case is done by applying the ν rule with \perp as the invariant, the two subderivations being built by induction.

Contraction is also an instance of the ν rule, choosing $(\lambda x. \nu Bx \wp \nu Bx)$ as the co-invariant. The proof of co-invariance follows from that of $(B(\nu B \wp \nu B)x, (B\nu Bx \wp B\nu Bx)^\perp)$, which in turn is a particular case of the more general form of derivation that we are going to build:

$$\vdash A(\nu B_1 \wp \nu B_1) \dots (\nu B_n \wp \nu B_n)x, (A\nu B_1 \dots \nu B_nx \wp A\nu B_1 \dots \nu B_nx)^\perp$$

where A is a fully asynchronous n -ary monotonic context. We prove this by induction on A .

- It is trivial if A is an inequality, \top or \perp .
- A is $\lambda p_1 \dots \lambda p_n \lambda x. p_i x$: we have to prove $(\nu B_i x \wp \nu B_i x, (\nu B_i x \wp \nu B_i x)^\perp)$ which is an instance of *init*.
- A is $A_1 \wp A_2$: we replace $(A(\nu B_i)_i x \wp A(\nu B_i)_i x)^\perp$ by the equivalent formula $(A_1(\nu B_i)_i x \wp A_1(\nu B_i)_i x)^\perp \otimes (A_2(\nu B_i)_i x \wp A_2(\nu B_i)_i x)^\perp$, thanks to *cut*. We can then operate the appropriate splitting allowing us to conclude by induction hypothesis on A_1 and A_2 .
- A is $A_1 \& A_2$: we build a derivation of $(A_1(\nu B_i \wp \nu B_i)_i x, (A(\nu B_i)_i x \wp A(\nu B_i)_i x)^\perp)$, by induction hypothesis on A_1 and a *deep* cut using the fact that:

$$(A_1(\nu B_i)_i)^\perp \multimap (A(\nu B_i)_i)^\perp$$

The corresponding derivation for B_2 is built the same way.

- A is $\forall x. A'x$: we introduce the quantifier, and instantiate the two existentials under the tensor as before, thanks to a cut.
- A is $\lambda p_1 \dots \lambda p_n. \nu(A'p_1 \dots p_n)$: we conclude by induction hypothesis on

$$\lambda p_1 \dots \lambda p_n \lambda p_{n+1}. A'p_1 \dots p_{n+1}$$

with $A'(\nu B_i)_{i \leq n}$ as B_{n+1} using the following widget:

$$Id \frac{\frac{A'(\nu B_i \wp \nu B_i)_i(\nu B_{n+1} \wp \nu B_{n+1})y, (A'(\nu B_i)_i(\nu B_{n+1})y \wp A'(\nu B_i)_i(\nu B_{n+1})y)^\perp}{A'(\nu B_i \wp \nu B_i)_i(\nu B_{n+1} \wp \nu B_{n+1})y, (\nu B_{n+1}y \wp \nu B_{n+1}y)^\perp} Trivial}{\nu(A'(\nu B_i \wp \nu B_i)_i)x, (\nu(A'(\nu B_i)_i)x \wp \nu(A'(\nu B_i)_i)x)^\perp} \nu \quad cut$$

Lemma 1. *If $\vdash \Gamma$ is derivable in μMALL^\equiv then $\vdash [\Gamma]$ is derivable in LL2 .*

Proof The validity of this encoding strongly relies on the monotonicity of definitions. Indeed, Sx will simulate correctly μBx only if it does not get negated by the body of the definition after an application of μ . This is more formally expressed by the following derivation using a slight variation on the deep cut rule:

$$\frac{\frac{\frac{\frac{\text{Trivial}}{!(\forall y . [B]S y \multimap S y), \forall S . !(\forall y . [B]S y \multimap S y) \multimap Sx \vdash Sx}}{[B][\mu B]t \vdash [B][\mu B]t}}{!(\forall y . [B]S y \multimap S y), [\mu B]x \vdash Sx}}{!(\forall y . [B]S y \multimap S y), [B][\mu B]t \vdash [B]S t}}{\text{deep}}}{!(\forall y . [B]S y \multimap S y), [B][\mu B]t \vdash S t}}{\vdash [B][\mu B]t \multimap [\mu B]t}$$

The μ rule is naturally given by the encoding. The last non-trivial rule is $\mu\nu$: it is translated by an instance of the identity, involving a second-order \forall introduction followed by the corresponding introduction of the existential. It is the only time a second-order existential is not instantiated by an $[\cdot]$ translation. \square

Lemma 2. *If there is a focused cut-free derivation of $\vdash [\Gamma]$, then there is a cut-free derivation of $\vdash \Gamma$.*

Definition 4. *Let Θ be a set of formulas. We define:*

$$\lfloor P \rfloor_\Theta = \begin{cases} \mu Bx & \text{if } P = Sx \text{ for } S \text{ atomic and } (\forall y . [B]S y \multimap S y)^\perp \in \Theta \\ Q & \text{if } [Q] = P \end{cases}$$

As usual we extend this notation to multisets. We'll also forget to specify what is Θ when it's obvious.

Proof The precise statement is:

1. If there is a proof of $\vdash \Theta : \Gamma \uparrow \Delta$ where $\Theta = \{(\forall y . [B_i]S_i y \multimap S_i y)^\perp \mid i \in I\}$, and Γ, Δ are multisets of encodings or (positive) instances of an S_i , then there is a proof of $\vdash [\Gamma, \Delta]$.
2. If there is a proof of $\vdash \Theta : \Gamma \Downarrow P$, with the same condition on Θ then there is a proof of $\vdash [P]$.

The asynchronous cases are easy. Only the introduction of second-order universal quantifier is not directly mapped to μMALL^\equiv . But it does not change the $[\cdot]$ translation.

Focusing on an unfolding hypothesis in Θ , the polarity of S makes it look exactly how we want:

$$\frac{\frac{\vdots}{\vdash \Theta : \Gamma \Downarrow [B_i]S_i y} \quad \frac{}{\vdash \Theta : S_i x \Downarrow (S_i x)^\perp}}{\vdash \Theta : \Gamma, S_i x \Downarrow [B_i]S_i x \otimes (S_i x)^\perp}}{\vdash \Theta : \Gamma, S_i x \Downarrow \exists y . [B_i]S_i y \otimes (S_i y)^\perp}$$

In the synchronous case, we now consider the introduction of second-order existential quantifier. Focusing on it introduces the existential and the tensor, but also

the exponential. The instantiated invariant is of the form $[I]$ for some closed I , because cut-elimination and focusing never change the instantiations. Thus, Θ is useless in the invariance subderivation. The derivation has the form:

$$\frac{\frac{\frac{\vdots}{\vdash \Theta : \uparrow \forall y. [BIy \multimap Iy]}{\vdash \Theta : \downarrow \forall y. [BIy \multimap Iy]} \quad \frac{\vdots}{\vdash \Theta : \Gamma \Downarrow [Ix]^\perp}}{\vdash \Theta : \Gamma \Downarrow (!\forall y. [B][I]y \multimap [I]y) \otimes [Ix]^\perp}}{\vdash \Theta : \Gamma \Downarrow \exists S. !(\forall y. [B]Sy \multimap Sy) \otimes (Sx)^\perp}$$

Which does translate well into:

$$\frac{\frac{\frac{\vdots}{\vdash [I], (Ix)^\perp} \quad \frac{\frac{\vdots}{\vdash BIy \multimap Iy}}{\vdash \overline{BI}^\perp y, Iy}}{\vdash [I], \overline{Bx}}}{\vdash [I], (\mu Bx)^\perp} =$$

Finally, the existential can be instantiated by its S_i . The $\mu\nu$ rule will be used to decode the derivation, which has the following form:

$$\frac{\frac{\frac{\frac{\vdots}{\vdash \Theta : \uparrow \forall y. [B]Iy \multimap Iy}}{\vdash \Theta : \downarrow \forall y. [B]Iy \multimap Iy} \quad \frac{\vdots}{\vdash \Theta : Sx \Downarrow (Sx)^\perp}}{\vdash \Theta : Sx \Downarrow (!\forall y. [B]Sy \multimap Sy) \otimes (Sx)^\perp}}{\vdash \Theta : Sx \Downarrow \exists S. !(\forall y. [B]Sy \multimap Sy) \otimes (Sx)^\perp}$$

□

As shown in the above proof, fixed points can be encoded by means of second-order quantification and exponentials. However, first-order MALL with exponentials and first-order MALL with fixed points are incomparable.

It has been observed [Gir92,SH93] that exponentials and non-monotonic definitions combine to yield inconsistency: for example, the definition $p \equiv p^\perp$ (that is, the fixed point $\mu\lambda p. p^\perp$) does not lead to an inconsistency, whereas the definition $p \equiv ?(p^\perp)$ (that is, $\mu\lambda p. ?(p^\perp)$) does. To reproduce the latter inconsistency in μMALL^\perp , one needs to be able to unfold the expression $\nu\lambda p. !(p^\perp)$. But this is not implied by Proposition 1 since its body is not monotonic. Thus, even in presence of exponentials, we currently do not have any example of non-monotonic definition that invalidates the consistency of μMALL^\perp .

3 Focused proofs

As we have explained in the introduction, completeness of a focused proof system is a valuable property for a logic to possess. Focused proofs have applications in proof-

search since it reduces the proof-search space by limiting the situations when backtracking is necessary. Focused proofs are also useful for justifying game theoretic semantics [MS05] and have been central to the design of Ludics [Gir01].

A good focused proof system for μMALL^\equiv is not a simple consequence of the translation of fixed points into LL2 that is used in the proof of Theorem 1: applying linear logic focusing to the result of that translation leads to a poorly structured system that is not consistent with our classification of connectives as asynchronous and synchronous. On the contrary, we present the proof system in Figure 2 as a good candidate for a focused proof system for μMALL^\equiv . We use explicit annotations of the sequents in the style of Andreoli. In the synchronous phase sequents have the form $\vdash \Gamma \Downarrow P$. In the asynchronous phase they have the form $\vdash \Gamma \Uparrow \Delta$ where Γ and Δ are both multisets of formulas. In both sequents, Γ is a multiset of synchronous formulas and ν -expressions. The convention on Δ is a slight departure from Andreoli's original proof system where Δ is a list (which can be used to provide a fixed but arbitrary ordering of the asynchronous phase).

Asynchronous phase	Synchronous phase
$\frac{\vdash \Gamma \Uparrow P, Q, \Delta}{\vdash \Gamma \Uparrow P \wp Q, \Delta} \quad \frac{\vdash \Gamma \Uparrow P, \Delta \quad \vdash \Gamma \Uparrow Q, \Delta}{\vdash \Gamma \Uparrow P \& Q, \Delta}$	$\frac{\vdash \Gamma \Downarrow P \quad \vdash \Gamma' \Downarrow Q}{\vdash \Gamma, \Gamma' \Downarrow P \otimes Q} \quad \frac{\vdash \Gamma \Downarrow P_i}{\vdash \Gamma \Downarrow P_0 \oplus P_1}$
$\frac{\vdash \Gamma \Uparrow \Delta}{\vdash \Gamma \Uparrow \perp, \Delta} \quad \frac{}{\vdash \Gamma \Uparrow \top, \Delta} \quad \frac{\{\vdash \Gamma \theta \Uparrow \Delta \theta : \theta \in \text{csu}(s \doteq t)\}}{\vdash \Gamma \Uparrow s \neq t, \Delta}$	$\frac{}{\vdash \Downarrow \mathbf{1}} \quad \frac{}{\vdash \Downarrow t = t}$
$\frac{\vdash \Gamma \Uparrow Pc, \Delta}{\vdash \Gamma \Uparrow \forall x. Px, \Delta} \quad c \text{ new}$	$\frac{\vdash \Gamma \Downarrow Pt}{\vdash \Gamma \Downarrow \exists x. Px}$
$\frac{\vdash \Gamma \Uparrow St, \Delta \quad \vdash \Uparrow BSx, Sx^\perp}{\vdash \Gamma \Uparrow \nu Bt, \Delta} \quad x \text{ new} \quad \frac{\vdash \Gamma, \nu Bt \Uparrow \Delta}{\vdash \Gamma \Uparrow \nu Bt, \Delta}$	$\frac{\vdash \Gamma \Downarrow B(\mu B)x}{\vdash \Gamma \Downarrow \mu Bx} \quad \frac{}{\vdash \nu \bar{B}x \Downarrow \mu Bx}$
Switching (where P is synchronous, Q asynchronous)	
$\frac{\vdash \Gamma, P \Uparrow \Delta \quad \vdash \Gamma \Downarrow P \quad \vdash \Gamma \Uparrow Q}{\vdash \Gamma \Uparrow P, \Delta \quad \vdash \Gamma, P \Uparrow \quad \vdash \Gamma \Downarrow Q}$	

Fig. 2. A focused proof-system for μMALL^\equiv

The rules for equality are not surprising. The main novelty here is the treatment of fixed points. Depending on the body, both μ and ν rules can be applied any number of times — but not with any co-invariant concerning ν . Notice for example that an instance of $\mu\nu$ can be η -expanded into a larger derivation, unfolding both fixed points to apply $\mu\nu$ on the recursive occurrences. As a result, each of the fixed point connectives has two rules in the focused system: one treats it as “an atom” and the other one as an expression with “internal structure.”

In accord with Definition 2, μ is treated during the synchronous phase and ν during the asynchronous phase. (Alternatives to this choice are discussed later.) Roughly, what the focused system implies is that if a proof involving a ν -expression proceeds by co-

induction on it, then this co-induction can be done at the beginning; otherwise that formula can be ignored in the whole derivation, except for the $\mu\nu$ rule. Focusing on a μ -expression yields two choices: unfolding or applying the initial rule for fixed points. If the body is fully synchronous, the focusing will never be lost. For example, if nat is the (fully synchronous) expression $\mu(\lambda nat.\lambda x. x = 0 \oplus \exists y.x = s y \otimes nat y)$, then focusing puts a lot of structure on a proof of $\Gamma \Downarrow nat t$: either t is a ground term representing a natural number and Γ is empty, or $t = s^n x$ for some $n \geq 0$ and Γ is $\{(nat x)^\perp\}$.

We shall now proceed with the completeness proof.

3.1 Trivial extra structure

We first slightly modify the system for technical reasons, without changing its expressivity. Both changes are quite obvious to apply as they only involve modifications of some leafs of a derivation.

- We add a new fixed point constructor ν_0 . The ν rule applies only on ν and the initial rule applies only on ν_0 . The connective ν_0 will not be classified as an asynchronous or synchronous, it is *something else*.

This new connective is only about marking some greatest fixed points as “frozen”: one can’t use ν on them. To make things clear we should also introduce a μ_0 which behaves exactly as μ , and is also classified as synchronous. This allows us to extend the negation:

$$(\nu_0 Bt)^\perp \stackrel{def}{=} \mu_0 \overline{Bt}$$

It is possible to derive ν from ν_0 using the ν rule:

$$\frac{\frac{\overline{\vdash B\nu_0 Bt, \overline{B\mu_0 \overline{Bt}}}}{\vdash \Gamma, \nu_0 Bt} \quad \overline{\vdash B\nu_0 Bt, \mu_0 \overline{Bt}} \quad \text{init}}{\vdash \Gamma, \nu Bt}$$

And that’s indeed what should be done before applying the initial rule.

- We also avoid that the initial asynchronous rules can be applied before other asynchronous rules. It means that the rules \top and \neq only apply when there is no asynchronous formula in the context. It makes it possible to apply any applicable asynchronous rule first without increasing the size of a derivation.

This “preprocessing” is only a technical device for the focalization proof, we stress that it is possible in the focused system to apply \top before other asynchronous rules, and that the initial rule does not have to be expanded.

Definition 5. *In this slightly revised system, two measures become interesting: $h_\mu(\Pi)$ and $|\Pi|$, which are both ordinals. $|\Pi|$ is the number of connectives in the conclusion, counting 1 for μ and ν expressions, but 0 for ν_0 . The μ -height of a derivation Π with subderivations $(\Pi_i)_i$ is inductively defined by:*

$$h_\mu(\Pi) = \begin{cases} 1 + \sup\{h_\mu(\Pi_i)\} & \text{if the first rule of } \Pi \text{ is } \mu \text{ or } \nu; \\ \sup\{h_\mu(\Pi_i)\} & \text{otherwise.} \end{cases}$$

Proposition 4. *The lexicographic order on $(h_\mu(\Pi), |\Pi|)$ is compatible with the sub-derivation order.*

Proof Any application removes one connective and thus decreases $|\Pi|$ (without changing $h_\mu(\Pi)$), except μ and ν which decrease $h_\mu(\Pi)$. \square

The couple $(h_\mu(\Pi), |\Pi|)$ will be simply called the *measure* in the followings.

3.2 Preliminaries

Lemma 3. *Proofs support instantiation: if σ ranges over first-order variables and $\vdash \Gamma$ then $\vdash \Gamma\sigma$. Moreover, the instantiated derivation has a least or equal measure.*

Proof This property is a standard and straightforward one, as the fixed points do not change anything here. \square

Lemma 4. *If S and S' are both covariants for B then so is $S \oplus S'$. Moreover the resulting covariance proof has the same $h_\mu()$ height as the highest original proof of invariance.*

Proof The proof of invariance of $S \oplus S'$ starts with a $\&$. Then we get the proof of $B(S \oplus S')\mathbf{x}, S\mathbf{x}^\perp$ (resp. $B(S \oplus S')\mathbf{x}, S'\mathbf{x}^\perp$) from the proof of $BS\mathbf{x}, S\mathbf{x}^\perp$ (resp. $BS'\mathbf{x}, S'\mathbf{x}^\perp$). The transformation is straightforward and relies on monotonicity, and obviously does not increase the μ -height. \square

We now present some interesting notions introduced by Alexis Saurin [MS07], which make the focalization proof clear and simple.

Definition 6 (Descendant). *We distinguish several occurrences of a formula in a sequent, and in a derivation. We define the notion of immediate descendant between formulas involved in a rule application. For example in*

$$\frac{\vdash \Gamma, P \quad \vdash Q, \Delta}{\vdash \Gamma, P \otimes Q, \Delta}$$

P and Q are immediate descendants of $P \otimes Q$, and every formula in Δ, Γ in the premises is an immediate descendant of the corresponding formula in the conclusion. In the ν rule, the formulas from the co-invariance proofs are not descendants of any formula, and $S\mathbf{t}$ is the immediate descendant of $\nu B\mathbf{t}$ ¹. The relation of descentance is the reflexive and transitive closure of the immediate descentance.

Definition 7. *The positive trunk of a derivation is its largest open sub-derivation which contains only applications of synchronous rules.*

Definition 8. *We define the relation $<$ on the formulas of the base sequent of a derivation Π . $P < Q$ iff there exists P' , asynchronous descendant of P in Π , and Q' , synchronous descendant of Q , such that P' and Q' occur in the same sequent of the positive trunk of Π .*

¹ We shall only use that notion in positive trunks anyway.

The intended meaning of $P < Q$ is that we must focus on P before Q . Therefore, the natural question is the existence of minimal elements for that relation, equivalent to its acyclicity.

Proposition 5. *If Π starts with a synchronous rule, and P is minimal for $<$ in Π , then so are its descendants in their respective subderivations.*

Proof It is enough to notice how the $<$ relation evolves in a positive trunk. The relations below and on top of a \oplus rule are isomorphic. The same thing holds for \exists and μ . The application of $=$ and $\mathbf{1}$ ends the derivation, and hence the positive trunk. There only remains the interesting case: the tensor. In that case the relation below the tensor is (isomorphic to) the union of the two relations on top of it, in which only two points get merged, namely the two descendants of the split tensor. \square

Lemma 5. *The relation $<$ is acyclic.*

Proof We re-use previous observations on the evolution of $<$ in a proof and proceed by induction. There is not anything to do for the $=$ and $\mathbf{1}$ cases. If the derivation starts with a \oplus , \exists or μ , the acyclicity on $<$ on the conclusion comes from the acyclicity for the subderivation. For \otimes , assuming the acyclicity of $<$ on the premises, we cannot have a cycle in the conclusion: this cycle cannot lie within the ancestors of a single branch, so it has to involve the split tensor, but then it would have to be involved twice because it is the only node linking the two ancestors components, and we contradict again the acyclicity of $<$ on the premises. \square

3.3 Permutation lemmas

Lemma 6. *If $\vdash \Gamma, P$ where P is an asynchronous formula, then there is a derivation where P is active in the conclusion, and it is has smaller or equal size than the original.*

Proof We proceed by induction on the height of the proof. If P is not active in the first rule, then by induction make it active in the immediate subderivations where it occurs. Then permute the first two rules. The resulting derivation will have the same conclusion, and it is also easily checked that it has no more ν and ν rules, so it will have equal (in most of the cases) or smaller size (e.g. in $*/\top$).

The MALL permutations are usual and interact well with our measure. Most of the permutations involving the new rules are not surprising, such as \otimes/ν :

$$\frac{\frac{\frac{\vdash \Gamma, P, S t \quad \vdash BS x, S x^\perp}{\vdash \Gamma, P, \nu B t} \quad \vdash \Gamma', P'}{\vdash \Gamma, \Gamma', P \otimes P', \nu B t}}{\vdash \Gamma, \Gamma', P \otimes P', \nu B t} \longrightarrow \frac{\frac{\frac{\vdash \Gamma, P, S t \quad \vdash \Gamma', P'}{\vdash \Gamma, \Gamma', P \otimes P', S t} \quad \vdash BS x, S x^\perp}{\vdash \Gamma, \Gamma', P \otimes P', \nu B t}}$$

& / ν holds thanks to Lemma 4:

$$\begin{array}{c}
\frac{\frac{\frac{\Pi}{\vdash \Gamma, P, S t} \quad \frac{\Pi_S}{\vdash BSx, Sx^\perp} \quad \frac{\Pi'}{\vdash \Gamma, P', S' t} \quad \frac{\Pi_{S'}}{\vdash BS'x, S'x^\perp}}{\vdash \Gamma, P, \nu Bt} \quad \frac{\quad}{\vdash \Gamma, P', \nu Bt}}{\vdash \Gamma, P \& P', \nu Bt} \\
\Downarrow \\
\frac{\frac{\frac{\frac{\Pi}{\vdash \Gamma, P, S t} \quad \frac{\Pi'}{\vdash \Gamma, P', S' t}}{\vdash \Gamma, P, S t \oplus S' t} \quad \frac{\frac{\phi_1(\Pi_S)}{\vdash B(S \oplus S')x, (Sx)^\perp} \quad \frac{\phi_2(\Pi_{S'})}{\vdash B(S \oplus S')x, (S'x)^\perp}}{\vdash \Gamma, P \& P', S t \oplus S' t} \quad \frac{\quad}{\vdash B(S \oplus S')x, ((S \oplus S')x)^\perp}}{\vdash \Gamma, P \& P', \nu Bt}}{\vdash \Gamma, P \& P', \nu Bt}} \&
\end{array}$$

Another non-trivial case is \otimes / \neq which makes use of Lemma 3:

$$\frac{\left\{ \frac{\vdash (\Gamma, P)\sigma \mid \sigma \in csu(u, v)}{\vdash \Gamma, P, (u = v)^\perp} \quad \vdash \Gamma', Q \right\}}{\vdash \Gamma, \Gamma', P \otimes Q, (u = v)^\perp} \rightarrow \left\{ \frac{\left\{ \frac{\vdash (\Gamma, P)\sigma \quad \vdash (\Gamma', Q)\sigma}{\vdash (\Gamma, \Gamma', P \otimes Q)\sigma} \mid \sigma \in csu(u, v) \right\}}{\vdash \Gamma, \Gamma', P \otimes Q, (u = v)^\perp} \right\}$$

□

Lemma 7. *In a sequent without any asynchronous formula, if P is a minimal synchronous then it can be applied first. Moreover the new derivation has a smaller or equal measure.*

Proof If P is active, there is nothing to do. Otherwise, we will make it active in the subderivations. We can do that by induction: by minimality of P the subderivation's conclusion does not have any asynchronous formula; and P is still minimal in the subderivations. Then we permute the first two layers of rules, which are synchronous. The permutation of synchronous rules are already know for MALL, and the new cases involving $=$ or μ are straightforward. The size preservation is easy to check. □

Theorem 2. *The focusing system is sound and complete with respect to μMALL^\perp .*

Proof Soundness is trivial. We prove completeness by induction:

- If there are any, pick an asynchronous formula *arbitrarily*, and transform the derivation by making that formula active thanks to Lemma 6. By induction, focalize the subderivations, and add the first rule in the focalized system.
- When there is no asynchronous formula left, we've shown in Lemma 5 that there is a minimal synchronous. Lemma 7 allow us to get a proof where this synchronous is active, we focalize its subderivations, choosing the minimal formula's subformulas as new minimal formulas, which makes it possible to glue that in the focalized system.

We actually still have a derivation in $\mu\text{MALL}^\perp + \nu_0$, but it is structured in a way that makes it simple to translate it to the focalized system. The only non-trivial thing is the handling of ν_0 : ν_0 is translated to ν and toplevel occurrences of ν_0 are moved to the left of \Downarrow . □

Remark 1. The derivations resulting from the transformation described here are not optimal. For example the immediate proof of $\vdash \nu Bt, \mu \overline{Bt}$ is first transformed into

$$\frac{\frac{\frac{\overline{\vdash \nu_0 Bt, \mu \overline{Bt}}}{\vdash \nu_0 Bt, \mu \overline{Bt}}}{\vdash \nu_0 Bt, \mu \overline{Bt}} \quad \frac{\frac{\vdash B\nu_0 Bt, \overline{B}\mu \overline{Bt}}{\vdash B\nu_0 Bt, \mu \overline{Bt}}}{\vdash \nu Bt, \mu \overline{Bt}}}{\vdash \nu Bt, \mu \overline{Bt}}$$

then it is focalized and finally translated in the focalized system. We end up with a derivation of the form

$$\frac{\frac{\frac{\overline{\vdash \nu Bt \Downarrow \mu \overline{Bt}}}{\vdash \nu Bt, \mu \overline{Bt} \Uparrow}}{\vdash \nu Bt \Uparrow \mu \overline{Bt}} \quad \frac{\vdash \nu Bt, \mu \overline{Bt}}{\vdash \Uparrow \nu Bt, \mu \overline{Bt}}}{\vdash \Uparrow \nu Bt, \mu \overline{Bt}} \quad \frac{\vdash \Uparrow B\nu Bt, \mu \overline{Bt}}{\vdash \Uparrow \nu Bt, \mu \overline{Bt}}$$

The point of the expansion of the initial rule was to allow the permutation of a $\&$ followed by ν on one side and $\mu\nu$ on the other.

4 Examples

We shall now give a few theorems in μMALL^\equiv . Although we do not give their derivations here, we stress that all of these examples are proved naturally in the focused proof system. The reader will also note that although μMALL^\equiv is linear, these derivations are intuitive and their structure resemble that of proofs in intuitionistic logic.

We first define a few least fixed points expressing basic properties of natural numbers. We assume two constants z and s of respective types n and $n \rightarrow n$. Note that all these definitions are fully synchronous.

$$\begin{aligned} \text{nat} &\stackrel{\text{def}}{=} \mu(\lambda \text{nat} \lambda x. x = z \oplus \exists y. x = s y \otimes \text{nat } y) \\ \text{even} &\stackrel{\text{def}}{=} \mu(\lambda \text{even} \lambda x. x = z \oplus \exists y. x = s (s y) \otimes \text{even } y) \\ \text{plus} &\stackrel{\text{def}}{=} \mu(\lambda \text{plus} \lambda a \lambda b \lambda c. a = z \otimes b = c \\ &\quad \oplus \exists a' \exists c'. a = s a' \otimes c = s c' \otimes \text{plus } a' b c') \\ \text{leq} &\stackrel{\text{def}}{=} \mu(\lambda \text{leq} \lambda x \lambda y. x = y \oplus \exists y'. y = s y' \otimes \text{leq } x y') \\ \text{half} &\stackrel{\text{def}}{=} \mu(\lambda \text{half} \lambda x \lambda h. (x = z \oplus x = s z) \otimes h = z \\ &\quad \oplus \exists x' \exists h'. x = s (s x') \otimes h = s h' \otimes \text{half } x' h') \end{aligned}$$

The following statements are theorems, all of which can be proved by induction. The main insights required for proving these theorems involve deciding which fixed point

expression should be introduced by induction: the proper invariant is not the difficult choice here since the context itself is adequate in these cases.

$$\begin{aligned}
&\vdash \forall x. \text{nat } x \multimap \text{even } x \oplus \text{even } (s \ x) \\
&\vdash \forall x. \text{nat } x \multimap \forall y \exists z. \text{plus } x \ y \ z \\
&\vdash \forall x. \text{nat } x \multimap \text{plus } x \ z \ x \\
&\vdash \forall x. \text{nat } x \multimap \forall y. \text{nat } y \multimap \forall z. \text{plus } x \ y \ z \multimap \text{nat } z
\end{aligned}$$

In the last theorem, the assumption $(\text{nat } x)^\perp$ is not needed and can be weakened, thanks to Proposition 3. In order to prove $(\forall x. \text{nat } x \multimap \exists h. \text{half } x \ h)$ one has to use a complete induction, *i.e.*, use the strengthened invariant $(\lambda x. \text{nat } x \otimes \forall y. \text{leq } y \ x \multimap \exists h. \text{half } y \ h)$.

A typical example of co-induction involves the simulation relation. Assume that $\text{step} : \text{state} \rightarrow \text{label} \rightarrow \text{state} \rightarrow o$ is an inductively defined relation encoding a labeled transition system. Simulation can be defined using the definition

$$\text{sim} \stackrel{\text{def}}{=} \nu(\lambda \text{sim} \lambda p \lambda q. \forall a \forall p'. \text{step } p \ a \ p' \multimap \exists q'. \text{step } q \ a \ q' \otimes \text{sim } p' \ q').$$

Reflexivity of simulation $(\forall p. \text{sim } p \ p)$ is proved easily by co-induction with the co-invariant $(\lambda p \lambda q. p = q)$. Instances of step are not subject to induction but are treated “as atoms”. Proving transitivity, that is,

$$\forall p \forall q \forall r. \text{sim } p \ q \multimap \text{sim } q \ r \multimap \text{sim } p \ r$$

is done by co-induction on $(\text{sim } p \ r)$ with the co-invariant $(\lambda p \lambda r. \exists q. \text{sim } p \ q \otimes \text{sim } q \ r)$. The focus is first put on $(\text{sim } p \ q)^\perp$, then on $(\text{sim } q \ r)^\perp$. The fixed points $(\text{sim } p' \ q')$ and $(\text{sim } q' \ r')$ appearing later in the proof are treated “as atoms”, as are all negative instances of step .

Except for the totality of half , all these theorems seem simple to prove using a limited number of heuristics. For example, one could first try to treat fixed points “as atoms”, an approach that would likely fail quickly if inappropriate. Second, depending on the “rigid” structure of the arguments to a fixed point expression, one might choose to either unfold the fixed point or attempt to use the surrounding context to generate an invariant.

5 Translating Intuitionistic Logic

The examples in the previous section make it clear that despite its simplicity and linearity, μMALL^\equiv can be related to a more conventional logic. In particular we are interested in drawing some connections with an extension of intuitionistic logic with inductive and coinductive definitions. We will show that the focusing of μMALL^\equiv derivations yields a similar result in the intuitionistic setting. A general approach for making such a connection is to first encode intuitionistic logic in μMALL^\equiv , focus the derivations of encodings, and translate them back to intuitionistic derivations. When doing so, it is interesting to minimize the use of exponentials in the encoding since these connectives weaken the focusing discipline. This is precisely what the extension of the asynchronous/synchronous classification allows. In the following, we show a simple first

step to this program, in which we actually capture a non-trivial fragment of intuitionistic logic extended with fixed points even though μMALL^\perp does not have exponentials at all.

We shall consider an intuitionistic logic in which there are no atomic formulas but where there are (positive) equalities and the two fixed point constructors μ and ν . Let μLJ^\perp be the proof system that extends Gentzen's cut-free LJ [Gen69] with the following rules for equality and (co)inductive expressions.

$$\frac{\{(G \vdash G)\theta : \theta \in \text{csu}(s \doteq t)\}}{\Gamma, s = t \vdash G} = L \quad \frac{}{\Gamma \vdash t = t} = R$$

$$\frac{BSx \vdash Sx \quad \Gamma, St \vdash G}{\Gamma, \mu Bt \vdash G} \mu L \quad \frac{}{\Gamma, \mu Bt \vdash \mu Bt} \mu_0 \quad \frac{\Gamma \vdash B(\mu B)t}{\Gamma \vdash \mu Bt} \mu R$$

$$\frac{\Gamma, B(\nu B)t \vdash G}{\Gamma, \nu Bt \vdash G} \nu L \quad \frac{}{\Gamma, \nu Bt \vdash \nu Bt} \nu_0 \quad \frac{Sx \vdash BSx \quad \Gamma \vdash St}{\Gamma \vdash \nu Bt} \nu R$$

We have observed (Prop. 3) that structural rules are admissible for fully asynchronous formulas of μMALL^\perp . This property will allow us to get a faithful encoding of a fragment of μLJ^\perp in μMALL^\perp despite the absence of exponentials. The encoding must be organized so that formulas appearing in the left-hand side of μLJ^\perp sequents must be encoded as fully asynchronous μMALL^\perp formulas. The only connectives allowed to appear negatively will thus be \wedge , \vee , $=$, μ and \exists . Moreover, the encoding must commute with negation, in order to translate the (co)induction rules correctly. This leaves no choice in the following design.

Definition 9. We restrict formulas to two fragments described by the two syntactic variables \mathcal{G} and \mathcal{H} :

$$\begin{aligned} \mathcal{G} ::= & \mathcal{G} \wedge \mathcal{G} \mid \mathcal{G} \vee \mathcal{G} \mid s = t \mid \mu(\lambda px. \mathcal{G} px)t \mid \exists x. \mathcal{G} x \\ & \mid \forall x. \mathcal{G} x \mid \mathcal{H} \supset \mathcal{G} \mid \nu(\lambda px. \mathcal{G} px)t \\ \mathcal{H} ::= & \mathcal{H} \wedge \mathcal{H} \mid \mathcal{H} \vee \mathcal{H} \mid s = t \mid \mu(\lambda px. \mathcal{H} px)t \mid \exists x. \mathcal{H} x \end{aligned}$$

Formulas in \mathcal{H} and \mathcal{G} are translated in μMALL^\perp as follows:

$$\begin{aligned} [P \wedge Q] &\stackrel{\text{def}}{=} [P] \otimes [Q] & [\forall x. Px] &\stackrel{\text{def}}{=} \forall x. [Px] \\ [P \vee Q] &\stackrel{\text{def}}{=} [P] \oplus [Q] & [\nu Bt] &\stackrel{\text{def}}{=} \nu [B]t \\ [s = t] &\stackrel{\text{def}}{=} s = t & [P \supset Q] &\stackrel{\text{def}}{=} [P] \multimap [Q] \\ [\mu Bt] &\stackrel{\text{def}}{=} \mu [B]t & [\lambda p \lambda x. B px] &\stackrel{\text{def}}{=} \lambda p \lambda x. [B px] \\ [\exists x. Px] &\stackrel{\text{def}}{=} \exists x. [Px] & & \end{aligned}$$

Proposition 6. For any $P \in \mathcal{G}$, P is provable in μLJ^\perp if and only if $[P]$ is provable in μMALL^\perp , under the restrictions that (co)invariants $\lambda x. Sx$ in μMALL^\perp (resp. μLJ^\perp) are such that Sx is in $[\mathcal{H}]$ (resp. \mathcal{H}).

Proof The proof transformations are simple and compositional. The induction rule is mapped to ν rule for $(\mu Bt)^\perp$; the left unfolding for co-inductives to μ for $(\nu Bt)^\perp$. In

order to restore the additive behavior of some intuitionistic rules (*e.g.*, $\wedge R$) and translate the structural rules, we can contract and weaken our fully asynchronous formulas on the left of μLJ^\perp sequents. \square

Linear logic provides an appealing proof theoretic setting because of its emphasis on dualities and on its clear separation of concepts (additive/multiplicative, asynchronous/synchronous). Our experience is that $\mu MALL^\perp$ is a good place to study focusing in the presence of least and greatest fixed point operators. To get similar results for intuitionistic logic, one can either work from scratch entirely within, say, μLJ^\perp , or use an encoding into linear logic. Given a mapping from intuitionistic to linear logic, and a complete focused proof system for linear logic, one can often build a complete “focalized” proof-system for intuitionistic logic. The usual encoding of intuitionistic logic into linear logic involves exponentials, which can damage focusing structures (by causing both synchronous and asynchronous phases to end). Hence, a careful study of the polarity of linear connectives must be done (cf. [DJS93,LM07]) in order to minimize the role played by the exponentials in such encodings. Here, as a result of Proposition 6, it is possible to get a complete focused system for μLJ^\perp on \mathcal{G} (under the assumptions that (co)invariants are in \mathcal{H}) that inherits the strong structure of the linear focusing derivations.

Although \mathcal{G} is not as expressive as full μLJ^\perp , it catches many interesting and useful problems. For example, any Horn-clause specification can be expressed in \mathcal{H} as a least fixed point and theorems that state properties such as totality or functionality of predicates defined in this manner are in \mathcal{G} . Theorems that state more model-checking properties, for example, $\forall x.p(x) \supset q(x)$, where p and q are one-placed least fixed point expressions over $[H]$, are also in \mathcal{G} . Finally, the theorems about natural numbers presented in Section 4 are within $[\mathcal{G}]$ although two of the derivations (for the totality of *half* and that the sum of natural numbers is a natural number) do not satisfy the restriction on co-invariants.

The logic μLJ^\perp is closely related to LINC [Tiu04]. The main difference is the absence of the ∇ quantifier in our system: we suspect that ∇ can be added to $\mu MALL^\perp$ in the same relatively orthogonal fashion that LINC added it to LJ. The resulting extension to $\mu MALL^\perp$ (and μLJ^\perp) should allow natural ways to reason about specifications involving variable bindings, in the manner illustrated in [BGM⁺07,Tiu04,Tiu05]. Another difference is that fixed points in LINC have to satisfy a stratification condition, which is strictly stronger than monotonicity; co-invariants also have to satisfy a technical restriction related to stratification. While our system, derived from linear logic, does not share such restrictions, neither difference is relevant when we restrict our attention to formulas in \mathcal{G} .

Interestingly, the fragment \mathcal{G} has already been identified in LINC [TNM05], and the Bedwyr system [BGM⁺07] implements a proof-search strategy for it that is complete under the assumption that all fixed points are noetherian (and hence that least and greatest fixed points coincide and that (co)induction can be restricted to unfolding). This strategy coincides with the focused system for μLJ^\perp restricted to noetherian fixed points: there is no need for any explicit contraction and you can always eagerly eliminate left-hand side (asynchronous) connectives before working on the goal (right-hand side); moreover there is no need for the initial rule $\mu\nu$.

6 Discussion about the focusing system

The design of the above focused proof system for μMALL^\equiv is rather satisfactory. For example, its treatment of μ as synchronous and ν as asynchronous is consistent with a similar treatment of these operators via game semantics given in [MS05,Sti96]. Focusing is also natural and helpful when trying to prove theorems in μMALL^\equiv , such as the examples proposed in Section 4. Finally, as we have seen in Section 5, this focused proof system yields another one for an intuitionistic logic similarly extended with fixed points, and accounts for the proof search strategy underlying the implemented prover Bedwyr [BGM⁺07]. It is worth noting, however, two unusual aspects of focused proofs in μMALL^\equiv .

6.1 A choice inside asynchronous rules.

As we noted, there are two rules for each of the fixed point connectives. Having a choice of rules in the asynchronous phase is, at first, rather surprising since it is during this phase of proof construction that we expect to see invertible rules and no choices. One way to look at this is that, in fact, the ν -connective should be *annotated* or divided into an infinite number of different connectives. In particular, consider replacing the ν constructor with both ν_ϵ (with the same types and arity as ν) and ν_S (where S is an annotated formula abstraction of the appropriate type). Now consider the proof system that results from replacing the three rules involving ν in Figure 2 by the rules

$$\frac{\frac{\vdash \Gamma \uparrow S t, \Delta \quad \vdash \uparrow BSx, Sx^\perp}{\vdash \Gamma \uparrow \nu_S Bt, \Delta} \quad x \text{ new} \quad \frac{\vdash \Gamma, \nu_\epsilon Bx \uparrow \Delta}{\vdash \Gamma \uparrow \nu_\epsilon Bx, \Delta} \quad \frac{}{\vdash \nu_\epsilon \bar{B}x \downarrow \mu Bx}}$$

Notice that using such annotated formulas, there is no longer any choice in the asynchronous phase. Furthermore, if in the expression $\nu_S B$ it is really the case that S is a co-invariant, *i.e.*, (BSx, Sx^\perp) is provable, then the first inference rule is invertible.

From a focused proof of F , it is possible to extract an annotation of F that is provable in the disambiguated focused system. This extraction requires the non-trivial composition of co-invariants in a manner similar to that used for the permutation of ν and $\&$. Such annotations might be useful for the partial automation of proof search involving induction and co-induction. For example, ν connectives could be labeled with partial information about what to do with the connective in the asynchronous phase: unfold, freeze (*i.e.*, treat as atomic), use the sequent as the invariant, etc. Such hints might be enough to mechanize a large amount of simple but tedious proofs by (co)induction. Notice that since we have annotated ν but not μ , we should not think that ν 's with annotations are logical connectives: instead, such annotations hint at the structure of a particular proof involving that annotated expression.

6.2 Are the polarities of μ and ν forced?

While the classification of μ as synchronous and ν as asynchronous is rather satisfying and is backed by several other observations, that choice does not seem to be forced from the focusing point of view alone. Maybe μ can be handled in the asynchronous phase,

instead? After all the μ rule is invertible. Consider replacing the fixed point rules in the focused proof system in Figure 2 with the following four inference rules:

$$\frac{\vdash \Gamma \uparrow B(\mu B)t, \Delta}{\vdash \Gamma \uparrow \mu Bt, \Delta} \quad \frac{\vdash \Gamma, \mu Bt \uparrow \Delta}{\vdash \Gamma \uparrow \mu Bt, \Delta} \quad \frac{\vdash \Gamma \Downarrow S t \quad \vdash \uparrow BSx, (Sx)^\perp}{\vdash \Gamma \Downarrow \nu Bt} \quad \frac{}{\vdash \mu \bar{B}t \Downarrow \nu Bt}$$

We conjecture that the resulting proof system is complete for μMALL^\equiv . The non-trivial step in such a proof would involve the permuting of the inference rules for μ and $\&$. The invertibility of μ allows it, but we have not proved the termination of the whole transformation.

To go one step further, one wonders if arbitrary assignment of “bias” to expressions such as (μBt) and (νBt) can be made in a fashion similar to the way literals are given fixed but arbitrary “bias” in Andreoli’s original focused proof system [And92]. Thus, maybe some μ expressions can be synchronous while others are asynchronous.

7 Conclusion and Future Work

μMALL^\equiv is an elegant logic supporting reasoning on inductive and co-inductive specifications. We have shown that it has two important proof-theoretic properties: namely, cut-elimination and the completeness of focused proofs. The design and completeness of a focused proof system is the major contribution of this paper. We have also shown that μMALL^\equiv is expressive and formally connected it to a fragment of intuitionistic logic extended with fixed points, a step that brings μMALL^\equiv closer to applications. Finally, we have identified an implemented system that attempts to find focused proofs within the noetherian part of this logic.

There are a number of interesting open questions to consider next. At the proof theory level, we would like to understand better whether or not dropping the monotonicity requirement leads to inconsistency or not and to what extent we can provide alternative assignment of polarities (synchronous/asynchronous) to fixed points. We can also consider adding exponentials and atomic formulas to μMALL^\equiv so that all of μLJ^\equiv could be encoded (in which case, a precise connection to the focused proof systems of [LM07] should be explored). Such an extension to μMALL^\equiv could also be used to generalize the uses of induction in the linear logic programming setting of [PM05]. At the system designing and implementation level, our focused proof system should help in designing a logic engine that attempts to prove formulas involving induction and co-induction. Our hope is that the focused proof system would help in understanding the strengths and limitations of various heuristics for generating invariants and co-invariants.

Acknowledgments We thank Alexis Saurin for helpful discussions and the anonymous reviewers of a previous draft of this paper for their comments, which helped us to reorganize this paper. This work has been supported in part by INRIA through the “Equipes Associées” Slimmer and by the Information Society Technologies programme of the European Commission, Future and Emerging Technologies under the IST-2005-015905 MOBIUS project.

References

- [And92] Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *J. of Logic and Computation*, 2(3):297–347, 1992.
- [AP91] J.M. Andreoli and R. Pareschi. Linear objects: Logical processes with built-in inheritance. *New Generation Computing*, 9(3-4):445–473, 1991.
- [AvE82] K. R. Apt and M. H. van Emden. Contributions to the theory of logic programming. *J. of the ACM*, 29(3):841–862, 1982.
- [BGM⁺07] David Baelde, Andrew Gacek, Dale Miller, Gopalan Nadathur, and Alwen Tiu. The Bedwyr system for model checking over syntactic expressions. In Frank Pfenning, editor, *21th Conference on Automated Deduction*, number 4603 in LNAI, pages 391–397. Springer, 2007.
- [DJS93] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. The structure of exponentials: Uncovering the dynamics of linear logic proofs. In Georg Gottlob, Alexander Leitsch, and Daniele Mundici, editors, *Kurt Gödel Colloquium*, volume 713 of LNCS, pages 159–171. Springer, 1993.
- [Gen69] Gerhard Gentzen. Investigations into logical deductions. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, 1969.
- [Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
- [Gir92] Jean-Yves Girard. A fixpoint theorem in linear logic. An email posting to the mailing list linear@cs.stanford.edu, February 1992.
- [Gir93] Jean-Yves Girard. On the unity of logic. *Annals of Pure and Applied Logic*, 59:201–217, 1993.
- [Gir98] Jean-Yves Girard. Light linear logic. *Information and Computation*, 143, 1998.
- [Gir01] Jean-Yves Girard. Locus solum. *Mathematical Structures in Computer Science*, 11(3):301–506, June 2001.
- [HM94] Joshua Hodas and Dale Miller. Logic programming in a fragment of intuitionistic linear logic. *Information and Computation*, 110(2):327–365, 1994.
- [Laf04] Yves Lafont. Soft linear logic and polynomial time. *Theoretical Computer Science*, 318(1-2):163–180, 2004.
- [LM07] Chuck Liang and Dale Miller. Focusing and polarization in intuitionistic logic. In J. Duparc and T. A. Henzinger, editors, *CSL 2007: Computer Science Logic*, volume 4646 of LNCS, pages 451–465. Springer-Verlag, 2007.
- [Mil96] Dale Miller. Forum: A multiple-conclusion specification logic. *Theoretical Computer Science*, 165(1):201–232, September 1996.
- [MM00] Raymond McDowell and Dale Miller. Cut-elimination for a logic with definitions and induction. *Theoretical Computer Science*, 232:91–119, 2000.
- [MNPS91] Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.
- [MS05] Dale Miller and Alexis Saurin. A game semantics for proof search: Preliminary results. In *Proceedings of the Mathematical Foundations of Programming Semantics (MFPS)*, 2005.
- [MS07] Dale Miller and Alexis Saurin. From proofs to focused proofs: a modular proof of focalization in linear logic. In J. Duparc and T. A. Henzinger, editors, *CSL 2007: Computer Science Logic*, volume 4646 of LNCS, pages 405–419. Springer-Verlag, 2007.
- [MT03] Alberto Momigliano and Alwen Tiu. Induction and co-induction in sequent calculus. In Mario Coppo Stefano Berardi and Ferruccio Damiani, editors, *Post-proceedings of TYPES 2003*, number 3085 in LNCS, pages 293 – 308, January 2003.

- [PM05] Elaine Pimentel and Dale Miller. On the specification of sequent systems. In *LPAR 2005: 12th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, number 3835 in LNAI, pages 352–366, 2005.
- [SH93] Peter Schroeder-Heister. Rules of definitional reflection. In M. Vardi, editor, *Eighth Annual Symposium on Logic in Computer Science*, pages 222–232. IEEE Computer Society Press, IEEE, June 1993.
- [Sti96] Colin Stirling. Games for bisimulation and model checking. Notes for Mathfit Workshop on Finite Model Theory, University of Wales, Swansea, July 1996.
- [Tiu04] Alwen Tiu. *A Logical Framework for Reasoning about Logical Specifications*. PhD thesis, Pennsylvania State University, May 2004.
- [Tiu05] Alwen Tiu. Model checking for π -calculus using proof search. In Martín Abadi and Luca de Alfaro, editors, *CONCUR*, volume 3653 of *LNCS*, pages 36–50. Springer, 2005.
- [TNM05] Alwen Tiu, Gopalan Nadathur, and Dale Miller. Mixing finite success and finite failure in an automated prover. In *Proceedings of ESHOL'05: Empirically Successful Automated Reasoning in Higher-Order Logics*, pages 79 – 98, December 2005.