Least and greatest fixed points in linear logic

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$^=$, satisfies two fundamental proof theoretic properties. In particular, μMALL$^=$ 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$^=$ 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 SLDresolution [AvE82]; uniform proofs (goal-directed proofs) in intuitionistic and intuitionistic linear logics have been used to justify λProlog [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 $MALL^=$, can be reduced to deciding unification problems (under a mixed prefix) which is decidable for the first-order fragment of $MALL^=$. 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 postfixed 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 $MALL^=$ to $\mu\text{MALL}^=$ by adding the two fixed points μ and ν.

Besides considering fixed points as alternatives to the exponentials, there are other reasons for examining $\mu\text{MALL}^=$. 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 $\mu\text{MALL}^=$.

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 $\mu\text{MALL}^=$ 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 $\mu\text{MALL}^=$. In Section 4 we describe a few examples of (focused) derivations in $\mu\text{MALL}^=$. Section 5 shows how the proof theory of $\mu\text{MALL}^=$ 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 $\boldsymbol{s}, \boldsymbol{t}$; 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:

$$\begin{aligned} P ::= {} & P \otimes P \mid P \oplus P \mid P \parr P \mid P \,\&\, P \mid \mathbf{1} \mid \mathbf{0} \mid \bot \mid \top \\ \mid {} & \exists_\gamma x.Px \mid \forall_\gamma x.Px \mid s \stackrel{\gamma}{=} t \mid s \stackrel{\gamma}{\neq} t \mid \mu_{\gamma_1 \dots \gamma_n} B\boldsymbol{t} \mid \nu_{\gamma_1 \dots \gamma_n} B\boldsymbol{t} \end{aligned}$$

The syntactic variable γ ranges over all simple types that do not contain o. The quantifiers have type $(\gamma \to o) \to o$ and the equality and inequality have type $\gamma \to \gamma \to o$. The connectives μ and ν have type $(\tau \to \tau) \to \tau$ where τ is $\gamma_1 \to \cdots \to \gamma_n \to o$ 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 $\mu\text{MALL}^=$ grammar. We shall see in the following the advantages of using fixed points instead.

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

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

A body B is said to be monotonic *when for any variables p and $\boldsymbol{t}$, the negation normal and λ-normal form of $Bp\boldsymbol{t}$ does not contain any negated instance of p.*

MALL rules

$$\frac{}{\vdash \mathbf{1}} \quad \frac{\vdash \Gamma, P \quad \vdash \Delta, Q}{\vdash \Gamma, \Delta, P \otimes Q} \quad \frac{\vdash \Gamma, P, Q}{\vdash \Gamma, P \parr Q} \quad \frac{\vdash \Gamma}{\vdash \Gamma, \bot}$$

$$\frac{}{\vdash \Delta, \top} \quad \frac{\vdash \Gamma, P \quad \vdash \Gamma, Q}{\vdash \Gamma, P \mathbin{\&} Q} \quad \frac{\vdash \Gamma, P_i}{\vdash \Gamma, P_0 \oplus P_1}$$

First-order structure

$$\frac{\vdash \Gamma, Pt}{\vdash \Gamma, \exists x.Px} \quad \frac{\vdash \Gamma, Pc}{\vdash \Gamma, \forall x.Px}\ c \text{ new}$$

$$\frac{}{\vdash t = t} \quad \frac{\{\vdash \Gamma\theta : \theta \in csu(s \doteq t)\}}{\vdash \Gamma, s \neq t}$$

Fixed points (where S is closed, $\mathbf{x}$ is new)

$$\frac{\vdash \Gamma, B(\mu B)\mathbf{t}}{\vdash \Gamma, \mu B\mathbf{t}}\ \mu \qquad \frac{\vdash \Gamma, S\,\mathbf{t} \quad \vdash BS\,\mathbf{x}, (S\,\mathbf{x})^\perp}{\vdash \Gamma, \nu B\mathbf{t}}\ \nu \qquad \frac{}{\vdash \mu B\mathbf{t}, \nu\overline{B}\mathbf{t}}\ \mu\nu$$

Fig. 1. Inference rules for μMALL$^=$

We shall assume that *all bodies are monotonic*. In other words, negation ($\bullet^\perp$ for formulas and $\overline{\bullet}$ 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$^=$.

We present the inference rules for μMALL$^=$ 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 $\gamma_1 \to \cdots \to \gamma_n \to o$. 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:*

$$\frac{}{\vdash P, P^\perp}\ init \qquad \frac{\vdash \Gamma, B(\nu B)\mathbf{t}}{\vdash \Gamma, \nu B\mathbf{t}}\ \nu R$$

These results are standard, cf. [Tiu04]. The proof of the second one relies on monotonicity and is obtained by applying the ν rule with $B(\nu B)$ as the co-invariant.

Definition 2. *We classify as* asynchronous *(resp.* synchronous*) the connectives* $\parr$, $\bot$, $\&$, $\top$, $\forall$, $\neq$, ν *(resp.* $\otimes$, $\mathbf{1}$, $\oplus$, $\mathbf{0}$, $\exists$, $=$, μ*). 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 $\lambda p\lambda \boldsymbol{x}.Bp\boldsymbol{x}$ is said to be fully asynchronous (resp. fully synchronous) when the formula $Bp\boldsymbol{x}$ is fully asynchronous (resp. fully synchronous).*

Notice, for example, that $\lambda p\lambda \boldsymbol{x}.p\boldsymbol{x}$ is fully asynchronous and fully synchronous.

Proposition 2. *The following structural rules are admissible provided that B is fully asynchronous:*

$$\frac{\vdash \Gamma, \nu B\boldsymbol{t}, \nu B\boldsymbol{t}}{\vdash \Gamma, \nu B\boldsymbol{t}}\ \nu C \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \nu B\boldsymbol{t}}\ \nu W$$

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

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

The proof of this proposition can be found in [BM07]. This property plays a central role in the focusing proof system presented in Section 3 and is crucial in Section 5 for our encoding of intuitionistic logic extended with least and greatest fixed points.

Example 1. Units can be represented by means of $=$ and $\neq$. Assuming that 2 and 3 are two distinct constants, then we have $2 = 2 \multimapboth \mathbf{1}$ and $2 = 3 \multimapboth \mathbf{0}$ (and hence $2 \neq 2 \multimapboth \bot$ and $2 \neq 3 \multimapboth \top$). Here, $P \multimapboth Q$ denotes $\vdash (P \multimap Q) \,\&\, (Q \multimap P)$ and $P \multimap Q$ denotes the formula $P^\perp \parr Q$.

Example 2. The μ (resp. ν) connective is meant to represent least (resp. greatest) fixed points. For example $\nu(\lambda p.p)$ is provable (take any provable formula as the co-invariant), while its dual $\mu(\lambda p.p)$ is not provable. More precisely: $\mu(\lambda p.p) \multimapboth \mathbf{0}$ and $\nu(\lambda p.p) \multimapboth \top$.

Example 3. The least fixed point, as expected, entails the greatest. The following is a proof of $\mu B\boldsymbol{t} \multimap \nu B\boldsymbol{t}$.

$$\dfrac{\dfrac{\dfrac{}{\vdash B(\mu B)\boldsymbol{x}, \overline{B}(\nu\overline{B})\boldsymbol{x}}\ init}{\vdash B(\mu B)\boldsymbol{x}, \nu\overline{B}\boldsymbol{x}}\ \nu R \qquad \dfrac{}{\vdash \mu B\boldsymbol{t}, \nu\overline{B}\boldsymbol{t}}\ \mu\nu}{\vdash \nu\overline{B}\boldsymbol{t}, \nu B\boldsymbol{t}}\ \nu \text{ on } \nu B\boldsymbol{t} \text{ with } S := \mu B$$

The greatest fixed point entails the least fixed point when the fixed points are noetherian, *i.e.*, all unfoldings of B and $\overline{B}$ terminate.

In this paper we are investigating how far one can go without the exponentials, getting the infinite behavior from the meaning of fixed points instead of modalities. If we were to add, however, the usual inference rules for exponentials, the resulting proof system would yield $\mu B\boldsymbol{t} \multimapboth\ !\,\mu B\boldsymbol{t}$ (and equivalently $?\,\nu\overline{B}\boldsymbol{t} \multimapboth \nu\overline{B}\boldsymbol{t}$) provided that B is fully synchronous. In the language of the Logic of Unity (LU) [Gir93], fully asynchronous (resp. fully synchronous) would be negative (resp. positive) or right-permeable (resp. left-permeable) formulas. Mixing synchronous and asynchronous connectives would yield a neutral formula.

We now outline the proof of cut-elimination. Although it is indirect and relies on cut-elimination for full second-order linear logic (LL2), this is still a syntactic proof of cut-elimination. It yields consistency of μMALL$^=$ as well as relative soundness and completeness with respect to LL2.

Theorem 1. *The logic μMALL$^=$ enjoys cut-elimination.*

Proof Our proof consists in first translating μMALL$^=$ formulas and proofs into full second-order linear logic derivations, which are then normalized and focused, and finally translated back to cut-free μMALL$^=$ derivations. Formally speaking, the previous work on proof normalization for LL2 does not include equality, but all the previous work on equality has shown that it has little role to play in normalization.

We first define the translation from first-order to second-order. The translation commutes with the connectives of MALL$^=$ and the negation, and is defined as follows on the least fixed points:

$$\lceil \mu B \boldsymbol{x} \rceil = \forall S\ .\ !(\forall \boldsymbol{y}\ .\ \lceil B \rceil S \boldsymbol{y} \multimap S \boldsymbol{y}) \multimap S \boldsymbol{x}$$

The corresponding transformation of proofs is straightforward, relying on the monotonicity of bodies. We get a proof where all second-order instantiations are either of the form $\lceil I \rceil$ (from ν rules) or second-order eigenvariables (from $\mu\nu$ rules). Cut-elimination and focusing never change these instantiations.

It is possible to normalize the resulting LL2 derivations, and then apply Andreoli's result to yield even more structured normal forms. (We shall temporarily assume that the reader is familiar with the focusing proof system in [And92]. A description of this kind of system may otherwise be found in Section 3.) Doing so, we get exactly the derivations we want for transforming them back to μMALL$^=$. For example, focusing on an unfolding hypothesis translates immediately to the μ rule:

$$\dfrac{\dfrac{\vdash \Theta : \Gamma \Downarrow \lceil B_i \rceil S_i \boldsymbol{y} \qquad \overline{\vdash \Theta : S_i \boldsymbol{x} \Downarrow (S_i \boldsymbol{x})^\perp}}{\vdash \Theta : \Gamma, S_i \boldsymbol{x} \Downarrow \lceil B_i \rceil S_i \boldsymbol{x} \otimes (S_i \boldsymbol{x})^\perp}}{\vdash \Theta : \Gamma, S_i \boldsymbol{x} \Downarrow \exists \boldsymbol{y}. \lceil B_i \rceil S_i \boldsymbol{y} \otimes (S_i \boldsymbol{y})^\perp}$$

Similarly, focusing on the translation of a ν gives us either an instance of the ν rule:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash \Theta :\Uparrow \lceil BI\boldsymbol{y} \multimap I\boldsymbol{y} \rceil}{\vdash \Theta :\Uparrow \forall \boldsymbol{y}. \lceil BI\boldsymbol{y} \multimap I\boldsymbol{y} \rceil}}{\vdash \Theta :\Downarrow ! \forall \boldsymbol{y}. \lceil BI\boldsymbol{y} \multimap I\boldsymbol{y} \rceil} \qquad \vdash \Theta : \Gamma \Downarrow \lceil I\boldsymbol{x} \rceil^\perp}{\vdash \Theta : \Gamma \Downarrow (! \forall \boldsymbol{y}. \lceil B \rceil \lceil I \rceil \boldsymbol{y} \multimap \lceil I \rceil \boldsymbol{y}) \otimes \lceil I\boldsymbol{x} \rceil^\perp}}{\vdash \Theta : \Gamma \Downarrow \exists S. !(\forall \boldsymbol{y}. \lceil B \rceil S \boldsymbol{y} \multimap S \boldsymbol{y}) \otimes (S \boldsymbol{x})^\perp}\ S := \lceil I \rceil$$

or an instance of $\mu\nu$ (the unfolding hypothesis for S is in Θ):

$$\dfrac{\dfrac{\dfrac{\vdots}{\vdash \Theta :\Downarrow ! \forall \boldsymbol{y}. \lceil B \rceil S \boldsymbol{y} \multimap S \boldsymbol{y}} \qquad \overline{\vdash \Theta : S \boldsymbol{x} \Downarrow (S \boldsymbol{x})^\perp}}{\vdash \Theta : S \boldsymbol{x} \Downarrow (! \forall \boldsymbol{y}. \lceil B \rceil S \boldsymbol{y} \multimap S \boldsymbol{y}) \otimes (S \boldsymbol{x})^\perp}}{\vdash \Theta : S \boldsymbol{x} \Downarrow \exists S. !(\forall \boldsymbol{y}. \lceil B \rceil S \boldsymbol{y} \multimap S \boldsymbol{y}) \otimes (S \boldsymbol{x})^\perp}$$

For a more detailed proof, see [BM07]. □

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$^=$, 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$^=$.

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$^=$ 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$^=$. 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).

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 $\{(nat\ x)^\perp\}$.

Asynchronous phase

$$\frac{\vdash \Gamma \Uparrow P, Q, \Delta}{\vdash \Gamma \Uparrow P \parr Q, \Delta} \quad \frac{\vdash \Gamma \Uparrow P, \Delta \quad \vdash \Gamma \Uparrow Q, \Delta}{\vdash \Gamma \Uparrow P \,\&\, Q, \Delta}$$

$$\frac{\vdash \Gamma \Uparrow \Delta}{\vdash \Gamma \Uparrow \bot, \Delta} \quad \frac{}{\vdash \Gamma \Uparrow \top, \Delta} \quad \frac{\{\vdash \Gamma\theta \Uparrow \Delta\theta : \theta \in csu(s \doteq t)\}}{\vdash \Gamma \Uparrow s \neq t, \Delta}$$

$$\frac{\vdash \Gamma \Uparrow Pc, \Delta}{\vdash \Gamma \Uparrow \forall x. Px, \Delta}\ c \text{ new}$$

$$\frac{\vdash \Gamma \Uparrow S\,\boldsymbol{t}, \Delta \quad \vdash\Uparrow BS\boldsymbol{x}, S\boldsymbol{x}^{\perp}}{\vdash \Gamma \Uparrow \nu B\boldsymbol{t}, \Delta}\ \boldsymbol{x} \text{ new} \quad \frac{\vdash \Gamma, \nu B\boldsymbol{t} \Uparrow \Delta}{\vdash \Gamma \Uparrow \nu B\boldsymbol{t}, \Delta}$$

Synchronous phase

$$\frac{\vdash \Gamma \Downarrow P \quad \vdash \Gamma' \Downarrow Q}{\vdash \Gamma, \Gamma' \Downarrow P \otimes Q} \quad \frac{\vdash \Gamma \Downarrow P_i}{\vdash \Gamma \Downarrow P_0 \oplus P_1}$$

$$\frac{}{\vdash\Downarrow \mathbf{1}} \quad \frac{}{\vdash\Downarrow t = t}$$

$$\frac{\vdash \Gamma \Downarrow Pt}{\vdash \Gamma \Downarrow \exists x. Px}$$

$$\frac{\vdash \Gamma \Downarrow B(\mu B)\boldsymbol{x}}{\vdash \Gamma \Downarrow \mu B\boldsymbol{x}} \quad \frac{}{\vdash \nu\overline{B}\boldsymbol{x} \Downarrow \mu B\boldsymbol{x}}$$

Switching (where P is synchronous, Q asynchronous)

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

Fig. 2. A focused proof-system for μMALL$^=$

Theorem 2. *The focused system is sound and complete with respect to μMALL$^=$.*

Proof Soundness is trivial. We only give an outline of the completeness proof: see [BM07] for more details. The proof is by (transfinite) induction on $(h_\mu(\Pi), |\Pi|)$ where $h_\mu(\Pi)$ is the height of Π in terms of fixed point rules, and $|\Pi|$ is the size of the derivation's conclusion. We first prove two permutation lemmas which preserve this measure: one shows that if there is any asynchronous formula in the conclusion, the proof can be transformed such that this formula is active in the conclusion; the other shows that when there is no more asynchronous in the conclusion, it is possible to focus on a synchronous if it is *maximal*. Finally we prove that there is always a maximal formula in such a sequent. The notion of maximality is due to Alexis Saurin [MS07] and is crucial to make the proof clear and simple.

It is worth pointing out, however, that there is a non-trivial permutation of & and ν in the first of these lemmas. This permutation, which requires the ability to sum co-invariants (a consequence of the monotonicity assumption on fixed point expressions) is illustrated in Figure 3. □

4 Examples

We shall now give a few theorems in μMALL$^=$. 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$^=$ 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 $n \to n$. Note that all these definitions are fully synchronous.

$$nat \stackrel{def}{=} \mu(\lambda nat \lambda x.\ x = z \oplus \exists y.\ x = s\ y \otimes nat\ y)$$

$$\dfrac{\dfrac{\dfrac{\Pi}{\vdash \Gamma, P, S\,\boldsymbol{t}} \quad \dfrac{\Pi_S}{\vdash BS\boldsymbol{x}, S\boldsymbol{x}^{\perp}}}{\vdash \Gamma, P, \nu B\boldsymbol{t}} \quad \dfrac{\dfrac{\Pi'}{\vdash \Gamma, P', S'\boldsymbol{t}} \quad \dfrac{\Pi_{S'}}{\vdash BS'\boldsymbol{x}, S'\boldsymbol{x}^{\perp}}}{\vdash \Gamma, P', \nu B\boldsymbol{t}}}{\vdash \Gamma, P \mathbin{\&} P', \nu B\boldsymbol{t}}$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Pi}{\vdash \Gamma, P, S\,\boldsymbol{t}}}{\vdash \Gamma, P, S\,\boldsymbol{t} \oplus S'\boldsymbol{t}} \quad \dfrac{\dfrac{\Pi'}{\vdash \Gamma, P', S'\boldsymbol{t}}}{\vdash \Gamma, P', S\,\boldsymbol{t} \oplus S'\boldsymbol{t}}}{\vdash \Gamma, P \mathbin{\&} P', S\,\boldsymbol{t} \oplus S'\boldsymbol{t}} \quad \dfrac{\dfrac{\phi_1(\Pi_S)}{\vdash B(S \oplus S')\boldsymbol{x}, (S\boldsymbol{x})^{\perp}} \quad \dfrac{\phi_2(\Pi_{S'})}{\vdash B(S \oplus S')\boldsymbol{x}, (S'\boldsymbol{x})^{\perp}}}{\vdash B(S \oplus S')\boldsymbol{x}, ((S \oplus S')\boldsymbol{x})^{\perp}}\ \&}{\vdash \Gamma, P \mathbin{\&} P', \nu B\boldsymbol{t}}$$

Fig. 3. The permutation of the & and the co-induction rules.

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

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.

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

In the last theorem, the assumption $(nat\ x)^{\perp}$ is not needed and can be weakened, thanks to Proposition 2. In order to prove $(\forall x.\ nat\ x \multimap \exists h.\ half\ x\ h)$ one has to use a complete induction, *i.e.*, use the strengthened invariant $(\lambda x.\ nat\ x \otimes \forall y.\ leq\ y\ x \multimap \exists h.\ half\ y\ h)$.

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

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

Reflexivity of simulation $(\forall p.\ sim\ p\ p)$ 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,

$$\forall p \forall q \forall r.\ sim\ p\ q \multimap sim\ q\ r \multimap sim\ p\ r$$

is done by co-induction on $(sim\ p\ r)$ with the co-invariant $(\lambda p \lambda r.\ \exists q.\ sim\ p\ q \otimes sim\ q\ r)$. The focus is first put on $(sim\ p\ q)^\perp$, then on $(sim\ q\ r)^\perp$. The fixed points $(sim\ p'\ q')$ and $(sim\ q'\ r')$ 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$^=$ 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$^=$ 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$^=$, 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$^=$ does not have exponentials at all.

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

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

$$\frac{BS\boldsymbol{x} \vdash S\boldsymbol{x} \quad \Gamma, S\boldsymbol{t} \vdash G}{\Gamma, \mu B\boldsymbol{t} \vdash G}\ \mu L \qquad \frac{}{\Gamma, \mu B\boldsymbol{t} \vdash \mu B\boldsymbol{t}}\ \mu_0 \qquad \frac{\Gamma \vdash B(\mu B)\boldsymbol{t}}{\Gamma \vdash \mu B\boldsymbol{t}}\ \mu R$$

$$\frac{\Gamma, B(\nu B)\boldsymbol{t} \vdash G}{\Gamma, \nu B\boldsymbol{t} \vdash G}\ \nu L \qquad \frac{}{\Gamma, \nu B\boldsymbol{t} \vdash \nu B\boldsymbol{t}}\ \nu_0 \qquad \frac{S\boldsymbol{x} \vdash BS\boldsymbol{x} \quad \Gamma \vdash S\boldsymbol{t}}{\Gamma \vdash \nu B\boldsymbol{t}}\ \nu R$$

We have observed (Prop. 2) that structural rules are admissible for fully asynchronous formulas of μMALL$^=$. This property will allow us to get a faithful encoding of a fragment of μLJ$^=$ in μMALL$^=$ despite the absence of exponentials. The encoding must be organized so that formulas appearing in the left-hand side of μLJ$^=$ sequents must be encoded as fully asynchronous μMALL$^=$ 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 3. *We restrict formulas to two fragments described by the two syntactic variables $\mathcal{G}$ and $\mathcal{H}$:*

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

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

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

Proposition 3. *For any $P \in \mathcal{G}$, P is provable in μLJ$^=$ if and only if $[P]$ is provable in μMALL$^=$, under the restrictions that (co)invariants $\lambda\boldsymbol{x}.S\,\boldsymbol{x}$ in μMALL$^=$ (resp. μLJ$^=$) are such that $S\,\boldsymbol{x}$ is in $[\mathcal{H}]$ (resp. $\mathcal{H}$).*

Proof The proof transformations are simple and compositional. The induction rule is mapped to ν rule for $(\mu B\boldsymbol{t})^\perp$; the left unfolding for co-inductives to μ for $(\nu B\boldsymbol{t})^\perp$. 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$^=$ sequents. □

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 μMALL$^=$ 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$^=$, 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 3, it is possible to get a complete focused system for μLJ$^=$ 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$^=$, 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$^=$ 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 μMALL$^=$ in the same relatively orthogonal fashion that LINC added it to LJ. The resulting extension to μMALL$^=$ (and μLJ$^=$) 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$^+$07] implements a proof-search strategy for it that is complete under the assumption that all fixed points are noetherian (and hence that least and greatest fixed points coincide and that (co)induction can be restricted to unfolding). This strategy coincides with the focused system for μLJ$^=$ 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$^=$ 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$^=$, 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$^+$07]. It is worth noting, however, two unusual aspects of focused proofs in μMALL$^=$.

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

$$\frac{\vdash \Gamma \Uparrow S\,\mathbf{t}, \Delta \quad \vdash \Uparrow BS\,\mathbf{x}, S\,\mathbf{x}^\perp}{\vdash \Gamma \Uparrow \nu_S B\mathbf{t}, \Delta}\ \mathbf{x}\text{ new} \qquad \frac{\vdash \Gamma, \nu_\epsilon B\mathbf{x} \Uparrow \Delta}{\vdash \Gamma \Uparrow \nu_\epsilon B\mathbf{x}, \Delta} \qquad \frac{}{\vdash \nu_\epsilon \overline{B}\mathbf{x} \Downarrow \mu B\mathbf{x}}$$

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.*, $(BS\,\boldsymbol{x}, S\,\boldsymbol{x}^\perp)$ 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:

$$\frac{\vdash \Gamma \Uparrow B(\mu B)\boldsymbol{t}, \Delta}{\vdash \Gamma \Uparrow \mu B\boldsymbol{t}, \Delta} \qquad \frac{\vdash \Gamma, \mu B\boldsymbol{t} \Uparrow \Delta}{\vdash \Gamma \Uparrow \mu B\boldsymbol{t}, \Delta} \qquad \frac{\vdash \Gamma \Downarrow S\,\boldsymbol{t} \quad \vdash \Uparrow BS\,\boldsymbol{x}, (S\,\boldsymbol{x})^\perp}{\vdash \Gamma \Downarrow \nu B\boldsymbol{t}} \qquad \frac{}{\vdash \mu\overline{B}\boldsymbol{t} \Downarrow \nu B\boldsymbol{t}}$$

We conjecture that the resulting proof system is complete for μMALL$^=$. 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 $(\mu B\boldsymbol{t})$ and $(\nu B\boldsymbol{t})$ 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$^=$ 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$^=$ is expressive and formally connected it to a fragment of intuitionistic logic extended with fixed points, a step that brings μMALL$^=$ 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$^=$ so that all of μLJ$^=$ could be encoded (in which case, a precise connection to the focused proof systems of [LM07] should be explored). Such an extension to μMALL$^=$ 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.

[BM07] David Baelde and Dale Miller. Least and greatest fixed points in linear logic: extended version. Technical report, available from the first author's web page, April 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.