Least and Greatest Fixed Points in Linear Logic

DAVID BAELDE, University of Minnesota

The first-order theory of MALL (multiplicative, additive linear logic) over only equalities is a well-structured 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: we establish weak normalization for it, and we design a focused proof system that we prove complete with respect to the initial system. That second result provides a strong normal form for cut-free proof structures that can be used, for example, to help automate proof search. We show how these foundations can be applied to intuitionistic logic.

1. INTRODUCTION

Inductive and coinductive definitions are ubiquitous in mathematics and computer science, from arithmetic to operational semantics and concurrency theory. These recursive definitions provide natural and very expressive ways to write specifications. The primary means of reasoning on inductive specifications is by induction, which involves the generalization of the tentative theorem in a way that makes it *invariant* under the considered inductive construction. Although the invariant might sometimes be the goal itself, it can be very different in general, sometimes involving concepts that are absent from the theorem statement. When proving theorems, most of the ingenuity actually goes into discovering invariants. Symmetrically, proving coinductive specifications is done by coinduction, involving *coinvariants* which again can have little to do with the initial specification. A proof theoretical framework supporting (co)inductive definitions can be used as a foundation for prototyping, model checking, and reasoning about many useful computational systems. But that great expressive power comes

with several difficulties such as undecidability, and even *nonanalyticity*: because of (co)induction rules and their arbitrary (co)invariants, proofs do not enjoy any reasonable form of subformula property. Nevertheless, we shall see that modern proof theory provides useful tools for understanding least and greatest fixed points and controlling the structure of proofs involving those concepts.

Arguably, the most important property of a logic is its consistency. In sequent calculus, consistency is obtained from cut elimination, which requires a symmetry between one connective and its dual, or in other words between construction and elimination, conclusion and hypothesis. The notions of polarity and focusing are more recent in proof theory but their growing importance puts them on par with cut elimination. Focusing organizes proofs in stripes of *asynchronous* and *synchronous* rules, removing irrelevant interleavings and inducing a reading of the logic based on macroconnectives aggregating stripes of usual connectives. Focusing is useful to justify game theoretic semantics [Delande and Miller 2008; Delande et al. 2010; Miller and Saurin 2006] and has been central to the design of Ludics [Girard 2001]. From the viewpoint of proof search, focusing plays the essential role of reducing the space of the search for a cut-free proof, by identifying situations when backtracking is unnecessary. In logic programming, it plays the more demanding role of correlating the declarative meaning of a program with its operational meaning, given by proof search. Various computational systems have employed different focusing theorems: much of Prolog's design and implementations can be justified by the completeness of SLD resolution [Apt and van Emden 1982]; uniform proofs (goal-directed proofs) in intuitionistic and intuitionistic linear logics have been used to justify λProlog [Miller et al. 1991] and Lolli [Hodas and Miller 1994]; the classical linear logic programming languages LO [Andreoli and Pareschi 1991], Forum [Miller 1996] and the inverse method [Chaudhuri and Pfenning 2005] have used directly Andreoli's general focusing result [Andreoli 1992] for linear logic. In the presence of fixed points, proof search becomes particularly problematic since cut-free derivations are not analytic anymore. Many systems use various heuristics to restrict the search space, but these solutions lack a proof theoretical justification. In that setting, focusing becomes especially interesting, as it yields a restriction of the search space while preserving completeness. Although it does not provide a way to decide the undecidable, focusing brings an appreciable leap forward, pushing further the limit where proof theory and completeness leave place to heuristics.

In this article, we propose a fundamental proof-theoretic study of the notions of least and greatest fixed point. By considering fixed points as primitive notions rather than, for example, encodings in second-order logic, we shall obtain strong results about the structure of their proofs. We introduce the logic μMALL which extends the multiplicative and additive fragments of linear logic (MALL) with least and greatest fixed points and establish its two fundamental properties, that is, cut elimination and focusing. There are several reasons to consider linear logic. First, its classical presentation allows us to internalize the duality between least and greatest fixed point operators, obtaining a simple, symmetric system. Linear logic also allows the independent study of fixed points and exponentials, two different approaches to infinity. Adding fixed points to linear logic without exponentials yields a system where they are the only source of infinity; we shall see that it is already very expressive. Finally, linear logic is simply a decomposition of intuitionistic and classical logics [Girard 1987]. Through this decomposition, the study of linear logic has brought a lot of insight to the structure of those more common systems. In that spirit, we provide in this paper some foundations that have already been used in more applied settings.

The logic μMALL was initially designed as an elementary system for studying the focusing of logics supporting (co)inductive definitions [Momigliano and Tiu 2003]; leaving aside the simpler underlying propositional layer (MALL instead of LJ), fixed points

are actually more expressive than this notion of definition since they can express mutually recursive definitions. But μMALL is also relatively close to type theoretical systems involving fixed points [Matthes 1999; Mendler 1991]. The main difference is that our logic is a first-order one, although the extension to second-order would be straightforward and the two fundamental results would extend smoothly. Inductive and coinductive definitions have also been approached by means of cyclic proof systems [Brotherston 2005; Santocanale 2001]. These systems are conceptually appealing, but generally weaker in a cut-free setting; some of our earlier work [Baelde 2009] addresses this issue in more details.

There is a dense cloud of work related to μMALL. Our logic and its focusing have been used to revisit the foundations of the system Bedwyr [Baelde et al. 2007], a proof search approach to model checking. A related work [Baelde 2009] carried out in μMALL establishes a completeness result for inclusions of finite automata leading to an extension of cyclic proofs. The treatment of fixed points in μMALL, as presented in this paper, can be used in full linear logic (μLL) and intuitionistic logic (μLJ). μLL has been used to encode and reason about various sequent calculi [Miller and Pimentel 2010]. μLJ has been given a game semantics [Clairambault 2009], and has been used in the interactive theorem prover Tac where focusing provides a foundation for automated (co)inductive theorem proving [Baelde et al. 2010], and in Nigam [2009] to extend a logical approach to tabling [Miller and Nigam 2007] where focusing is used to avoid redundancies in proofs. Finally, those logics have also been extended with (minimal) generic quantification [Baelde 2008b; Miller and Tiu 2005], which fully enables reasoning in presence of variable binding, *e.g.,* about operational semantics, logics or type systems.

The rest of this article is organized as follows. In Section 2, we introduce the logic, provide a few examples, and study its basic proof theory. Section 3 establishes cut elimination for μMALL, by adapting the candidates of reducibility argument to obtain a proof of weak normalization. Finally, we investigate the focusing of μMALL in Section 4, and present a simple application to intuitionistic logic.

2. μMALL

We assume some basic knowledge of simply typed λ-calculus [Barendregt 1992] which we leverage as a representation framework, following Church's approach to syntax. This allows us to consider syntax at a high-level, modulo $\alpha\beta\eta$-conversion. In this style, we write Px to denote a formula from which x has been totally abstracted out (x does not occur free in P), so that Pt corresponds to the substitution of x by t, and we write $\lambda x.P$ to denote a vacuous abstraction. *Formulas* are objects of type o, and the syntactic variable γ shall represent a term type, that is, any simple type that does not contain o. A *predicate* of arity n is an object of type $\gamma_1 \rightarrow \ldots \rightarrow \gamma_n \rightarrow o$, and a *predicate operator* (or simply *operator*) of first-order arity n and second-order arity m is an object of type $\tau_1 \rightarrow \ldots \rightarrow \tau_m \rightarrow \gamma_1 \rightarrow \ldots \rightarrow \gamma_n \rightarrow o$ where the τ_i are predicate types of arbitrary arity. We shall see that the term language can in fact be chosen quite freely: for example terms might be first-order, higher-order, or even dependently typed, as long as equality and substitution are defined.

We shall denote terms by s, t; formulas by P, Q; operators by A, B; term variables by x, y; predicate variables by p, q; and atoms (predicate constants) by a, b. The syntax of μMALL formulas is as follows:

$$P ::= P \otimes P \mid P \oplus P \mid P \,\mathscr{V}\, P \mid P \,\&\, P \mid \mathbf{1} \mid \mathbf{0} \mid \bot \mid \top \mid a\vec{t} \mid a^\perp \vec{t}$$
$$\mid\ \exists_\gamma x.\ Px \mid \forall_\gamma x.\ Px \mid s =_\gamma t \mid s \neq_\gamma t$$
$$\mid\ \mu_{\gamma_1 \ldots \gamma_n}(\lambda p \lambda \vec{x}.\ Pp\vec{x})\vec{t} \mid \nu_{\gamma_1 \ldots \gamma_n}(\lambda p \lambda \vec{x}.\ Pp\vec{x})\vec{t} \mid p\vec{t} \mid p^\perp \vec{t}$$

The quantifiers have type $(\gamma \to o) \to o$ and the equality and *disequality* (i.e., $\neq$) 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 the 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* (such as in $\nu(\lambda p.\ p \otimes \mu(\lambda q.\ \mathbf{1} \oplus a \otimes q))$, written $\nu p.\ p \otimes (\mu q.\ \mathbf{1} \oplus a \otimes q)$ for short) and *interleaved* (e.g., $\mu p.\ \mathbf{1} \oplus \mu q.\ \mathbf{1} \oplus p \otimes q$). Nested fixed points correspond to iterated (co)inductive definitions while interleaved fixed points correspond to mutually (co)inductive definitions, with the possibility of simultaneously defining an inductive and a coinductive.

Note that negation is not part of the syntax of our formulas, except for atoms and predicate variables. This is usual in classical frameworks, where negation is instead defined as an operation on formulas.

Definition 2.1 *Negation* $(P^\perp, \overline{B})$. *Negation* is the involutive operation on formulas satisfying the following equations:

$$
\begin{aligned}
(P \,\mathbin{⅋}\, Q)^\perp &\equiv P^\perp \otimes Q^\perp & (P \,\&\, Q)^\perp &\equiv P^\perp \oplus Q^\perp \\
\perp^\perp &\equiv \mathbf{1} & \top^\perp &\equiv \mathbf{0} \\
(s = t)^\perp &\equiv s \neq t & (\forall x.\ Px)^\perp &\equiv \exists x.\ (Px)^\perp \\
(\nu B \vec{t})^\perp &\equiv \mu \overline{B} \vec{t} & (a^\perp \vec{t})^\perp &\equiv a\vec{t} & (p^\perp \vec{t})^\perp &\equiv p\vec{t} \\
\overline{B} &\equiv \lambda p_1 \ldots \lambda p_m \lambda x_1 \ldots \lambda x_n.\ (B p_1^\perp \ldots p_m^\perp x_1 \ldots x_n)^\perp & &\text{for operators} \\
P^\perp &\equiv \lambda x_1 \ldots \lambda x_n.\ (Px_1 \ldots x_n)^\perp & &\text{for predicates}
\end{aligned}
$$

An operator B is said to be *monotonic* when it does not contain any occurrence of a negated predicate variable. We shall write $P \multimap Q$ for $P^\perp \,\mathbin{⅋}\, Q$, and $P \multimapboth Q$ for $(P \multimap Q) \,\&\, (Q \multimap P)$.

We shall assume that *all predicate operators are monotonic*, and do not have any free term variable. By doing so, we effectively exclude negated predicate variables $p^\perp$ from the logical syntax; they are only useful as intermediate devices when computing negations.

Example 2.2. We assume a type n and two constants 0 and s of respective types n and $n \to n$. The operator $(\lambda p \lambda x.\ x = 0 \oplus \exists y.\ x = s\ (s\ y) \otimes p\ y)$ whose least fixed point describes even numbers is monotonic, but $(\lambda p \lambda x.\ x = 0 \oplus \exists y.\ x = s\ y \otimes (p\ y \multimap \mathbf{0}))$ is nonmonotonic because of the occurrence of $p^\perp y$ that remains once the definition of $\multimap$ has been expanded and negations have been computed.

A *signature*, denoted by Σ, is a list of distinct typed variables. We write $\Sigma \vdash t : \gamma$ when t is a well-formed term of type γ under the signature Σ; we shall not detail how this standard judgment is derived. A *substitution* θ consists of a domain signature Σ, an image signature Σ', and a mapping from each $x : \gamma$ in Σ to some term t of type γ under Σ'. We shall denote the image signature Σ' by $\Sigma\theta$. Note that we do not require each variable from $\Sigma\theta$ to be used in the image of Σ: for example, we do consider the substitution from Σ to (Σ, x) mapping each variable in Σ to its counterpart in the extended signature. If $\Sigma \vdash t : \gamma$, then $t\theta$ denotes the result of substituting free variables in t by their image in θ, and we have $\Sigma\theta \vdash t\theta : \gamma$.

Our *sequents* have the form $\Sigma; \vdash \Gamma$ where the signature Σ denotes universally quantified terms,[1] and Γ is a multiset of formulas, that is, expressions of type o under Σ. Here, we shall make an exception to the higher-order abstract syntax notational convention: when we write $\Sigma; \vdash \Gamma$ using the metavariable Σ (i.e., without detailing the contents of the signature) we allow variables from Σ to occur in Γ. It is often important to distinguish different occurrences of a formula in a proof, or track a particular formula throughout a proof; such distinctions are required for a meaningful computational interpretation of cut elimination, and they also play an important role in our focusing mechanisms. In order to achieve this, we shall use the notion of *location*. From now on, we shall consider a formula not only as the structure that it denotes, namely an abstract syntax tree, but also as an address where this structure is located. Similarly, subformulas have their own locations, yielding a *tree of locations* and sublocations. We say that two locations are *disjoint* when they do not share any sublocation. Locations are independent of the term structure of formulas: all instantiations of a formula have the same location, which amounts to say that locations are attached to formulas abstracted over all terms. We shall not provide a formal definition of locations, which would be rather heavy, but a high-level description should give a sufficient understanding of the concept. A formal treatment of locations can be found in Girard [2001], and locations can also be thought of as denoting nodes in proof nets or variable names in proof terms. Locations allow us to make a distinction between *identical* formulas, which have the same location, and *isomorphic* formulas which only denote the same structure. When we talk of the *occurrences* of a formula in a proof, we refer to identical formulas occurring at different places in that derivation. We shall assume that formulas appearing in a sequent have *pairwise disjoint locations*. In other words, sequents are actually sets of formulas-with-location, which does not exclude that a sequent can contain several isomorphic formulas.

We present the inference rules for μMALL in Figure 1. Rules which are not in the identity group are called *logical rules*, and the only formula whose toplevel connective is required for the application of a logical rule is said to be *principal* in that rule application. In the $\neq$ rule, θ is a substitution of domain Σ ranging over universal variables, $\Gamma\theta$ is the result of applying that substitution to every term of every formula of Γ. In the ν rule, which provides both induction and coinduction, S is called the (co)invariant, and is a closed formula of the same type as νB, of the form $\gamma_1 \to \cdots \to \gamma_n \to o$. We shall adopt a proof search reading of derivations: for instance, we call the μ rule "unfolding" rather than "folding," and we view the rule whose conclusion is the conclusion of a derivation as the first rule of that derivation.

Inference rules should be read from the locative viewpoint, which we illustrate with a couple of key examples. In the $\forall$ and $\exists$ rules, the premise and conclusion sequents only differ in one location: the principal location is replaced by its only sublocation. The premise sequents of the $\neq$ rule are locatively identical to the conclusion sequent, except for the location of the principal $\neq$ formula that has been removed. Similarly in the $\&$ rule, the formulas of the context Γ are copied in the two premises, each of them occurring (identically) three times in the rule. In the axiom rule, the two formulas are locatively distinct but have dual structures. In the ν rule, the formulas from the coinvariance proofs have *new* locations, as well as the co-invariant in the main premise. This means that these locations can be changed at will, much like a renaming of bound variables. A greatest fixed point has infinitely many sublocations, regarding the coinvariants as its subformulas. In the μ rule, the formula $B(\mu B)\vec{t}$ is the only sublocation

---

[1] Term constants and atoms are viewed as being introduced, together with their types, in an external, toplevel signature that is never explicitly dealt with. Predicate variables are not found in either of those signatures; they cannot occur free in sequents.

Identity group

$$\frac{\Sigma;\vdash \Gamma, P^\perp \quad \Sigma;\vdash P, \Delta}{\Sigma;\vdash \Gamma, \Delta} \; cut \qquad \frac{}{\Sigma;\vdash P, P^\perp} \; init$$

MALL rules

$$\frac{\Sigma;\vdash \Gamma}{\Sigma;\vdash \Gamma, \perp} \; \perp \qquad \frac{\Sigma;\vdash \Gamma, P, Q}{\Sigma;\vdash \Gamma, P \,\invamp\, Q} \; \invamp \qquad \frac{\Sigma;\vdash \Gamma, P \quad \Sigma;\vdash \Delta, Q}{\Sigma;\vdash \Gamma, \Delta, P \otimes Q} \; \otimes \qquad \frac{}{\Sigma;\vdash \mathbf{1}} \; \mathbf{1}$$

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

First-order structure

$$\frac{\Sigma, x : \gamma;\vdash \Gamma, Px}{\Sigma;\vdash \Gamma, \forall_\gamma x.Px} \; \forall \qquad \frac{\Sigma \vdash t : \gamma \quad \Sigma;\vdash \Gamma, Pt}{\Sigma;\vdash \Gamma, \exists_\gamma x.Px} \; \exists$$

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

Fixed points

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

Fig. 1.   Inference rules for first-order μMALL.

of the least fixed point. Distinct occurrences of μB in $B(\mu B)$ (resp. νB in $B(\nu B)$) have distinct locations, so that the graph of locations remains a tree. It is easy to check that inference rules preserve the fact that sequents are made of disjoint locations.

Note that μMALL is a *conservative* extension of MALL, meaning that a MALL formula is derivable in MALL if and only if it is derivable in μMALL: it is easy to check that MALL and μMALL have the same cut-free derivations of MALL formulas, and cut elimination will allow us to conclude.

In the following, we use a couple of notational shortcuts. For conciseness, and when it does not create any ambiguity, we may use $\bullet$ to denote implicitly abstracted variables, for instance, $P\bullet x$ denotes $\lambda y.Pyx$. Similarly, we may omit abstractions, for instance, $\perp$ used as a coinvariant stands for $\lambda \vec{x}. \perp$ and, when S_1 and S_2 are predicates of the same type, $S_1 \,\invamp\, S_2$ stands for $\lambda \vec{x}. S_1\vec{x} \,\invamp\, S_2\vec{x}$. Finally, we shall omit the signature of sequents whenever unambiguous, simply writing $\vdash \Gamma$.

2.1 Equality

The treatment of equality dates back to Girard [1992] and Schroeder-Heister [1993], originating from logic programming. In the disequality rule, which is a case analysis on all unifiers, *csu* stands for *complete set of unifiers*, that is, a set $\mathcal{S}$ of unifiers of $u \doteq v$ such that any unifier σ can be written as $\theta\sigma'$ for $\theta \in \mathcal{S}$. For determinacy reasons, we assume a fixed mapping from unification problems to complete sets of unifiers, always taking $\{id\}$ for $csu(u \doteq u)$. Similarly, we shall need a fixed mapping from each unifier $\sigma' \in csu(u\theta \doteq v\theta)$ to a $\sigma \in csu(u \doteq v)$ such that $\theta\sigma' = \sigma\theta'$ for some θ'; existence is guaranteed since $\theta\sigma'$ is a unifier of $u \doteq v$. In the first-order case, and in general when most general unifiers exist, the *csu* can be restricted to having at most one element. But we do not rule out higher-order terms, for which unification is undecidable and complete sets of unifiers can be infinite [Huet 1975]; in implementations, we restrict to well-behaved fragments such as higher-order patterns [Miller 1992]. Hence, the left equality rule might be infinitely branching. But derivations remain inductive

structures (they don't have infinite branches) and are handled naturally in our proofs by means of (transfinite) structural induction. 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 do not make much assumptions on the language of terms.

It should be noted that our "free" equality is more powerful than the more usual Leibniz equality. Indeed, it implies the injectivity of constants: one can prove for example that $\forall x.\ 0 = s\ x \multimap \mathbf{0}$ since there is no unifier for $0 \doteq s\ x$. This example also highlights that constants and universal variables are two different things, since only universal variables are subject to unification, which is why we avoid calling them eigenvariables. It is also important to stress that the disequality rule does not and must not embody any assumption about the signature, just like the universal quantifier. That rule enumerates substitutions over open terms, not instantiations by closed terms. Otherwise, with an empty domain we would prove $\forall x.\ x = x \multimap \mathbf{0}$ (no possible instantiation for x) and $\forall x.\ x = x$, but not (without cut) $\forall x.\ \mathbf{0}$. Similarly, by considering a signature with a single constant $c : \tau_2$, so that τ_1 is empty while $\tau_1 \to \tau_2$ contains only $\lambda x.\ c$, we would indeed be able to prove $\forall x.\ x = x$ and $\forall x.\ x = x \multimap \exists y.\ x = \lambda a.\ y$ but not (without cut) $\forall x \exists y.\ x = \lambda a.\ y$.

Example 2.3. Units can be represented by means of = and $\neq$. Assuming that 2 and 3 are two distinct constants, then we have $2 = 2 \circ\!\!-\!\!\circ \mathbf{1}$ and $2 = 3 \circ\!\!-\!\!\circ \mathbf{0}$ (and hence $2 \neq 2 \circ\!\!-\!\!\circ \perp$ and $2 \neq 3 \circ\!\!-\!\!\circ \top$).

2.2 Fixed Points

Our treatment of fixed points follows from a line of work on definitions [Girard 1992; McDowell and Miller 2000; Momigliano and Tiu 2003; Schroeder-Heister 1993]. In order to make that lineage explicit and help the understanding of our rules, let us consider for a moment an intuitionistic framework (linear or not). In such a framework, the rules associated with least fixed points can be derived from Knaster-Tarski's characterization of an operator's least fixed point in complete lattices: it is the least of its pre-fixed points.[2]

$$\frac{\vec{x}; B S \vec{x} \vdash S \vec{x}}{\Sigma; \mu B \vec{t} \vdash S \vec{t}} \qquad \frac{\Sigma; \Gamma \vdash B(\mu B)\vec{t}}{\Sigma; \Gamma \vdash \mu B \vec{t}}$$

As we shall see, the computational interpretation of the left rule is recursion. Obviously, that computation cannot be performed without knowing the inductive structure on which it iterates. In other words, a cut on $S\vec{t}$ cannot be reduced until a cut on $\mu B\vec{t}$ is performed. As a result, a more complex left introduction rule is usually considered (e.g., in Momigliano and Tiu [2003]), which can be seen as embedding this suspended cut:

$$\frac{\Sigma; \Gamma, S\vec{t} \vdash P \quad \vec{x}; B S \vec{x} \vdash S \vec{x}}{\Sigma; \Gamma, \mu B \vec{t} \vdash P} \qquad \frac{\Sigma; \Gamma \vdash B(\mu B)\vec{t}}{\Sigma; \Gamma \vdash \mu B \vec{t}}$$

Notice, by the way, how the problem of suspended cuts (in the first set of rules) and the loss of subformula property (in the second one) relate to the arbitrariness of S, or in other words the difficulty of finding an invariant for proving $\Gamma, \mu B\vec{t} \vdash P$.

---

[2]Pre-fixed points of ϕ are those x such that $\phi(x) \leq x$.

Greatest fixed points can be described similarly as the greatest of the post-fixed points:

$$\frac{\Sigma; \Gamma, B(\nu B)\vec{t} \vdash P}{\Sigma; \Gamma, \nu B\vec{t} \vdash P} \qquad \frac{\Sigma; \Gamma \vdash S\vec{t} \quad \vec{x}; S\vec{x} \vdash BS\vec{x}}{\Sigma; \Gamma \vdash \nu B\vec{t}}$$

Example 2.4. Let B_{nat} be the operator $(\lambda N \lambda x.\; x = 0 \oplus \exists y.\; x = s\, y \otimes N\, y)$ and *nat* be its least fixed point μB_{nat}. Then the following inferences can be derived from the preceding rules.

$$\frac{\Sigma; \Gamma, S\, t \vdash P \quad \vdash S\, 0 \quad y; S\, y \vdash S\,(s\, y)}{\Sigma; \Gamma, nat\, t \vdash P} \qquad \frac{}{\Sigma; \Gamma \vdash nat\, 0} \qquad \frac{\Sigma; \Gamma \vdash nat\, t}{\Sigma; \Gamma \vdash nat\,(s\, t)}$$

Let us now consider the translation of those rules to classical linear logic, using the usual reading of $\Gamma \vdash P$ as $\vdash \Gamma^\perp, P$ where $(P_1, \ldots, P_n)^\perp$ is $(P_1^\perp, \ldots, P_n^\perp)$. It is easy to see that the above right introduction rule for μ (resp. ν) becomes the μ (resp. ν) rule of Figure 1, by taking $\Gamma^\perp$ for Γ. Because of the duality between least and greatest fixed points (i.e., $(\mu B)^\perp \equiv \nu \overline{B}$) the other rules collapse. The translation of the above left introduction rule for ν corresponds to an application of the μ rule of μMALL on $(\nu B\vec{t})^\perp \equiv \mu \overline{B}\vec{t}$. The translation of the left introduction rule for μ is as follows.

$$\frac{\vdash \Gamma^\perp, S^\perp\vec{t}, P \quad \vdash (BS\vec{x})^\perp, S\vec{x}}{\vdash \Gamma^\perp, (\mu B\vec{t})^\perp, P}$$

Without loss of generality, we can write S as $S'^\perp$. Then $(BS\vec{x})^\perp$ is simply $\overline{B}S'\vec{x}$ and we obtain exactly the ν rule of μMALL on $\nu\overline{B}$.

$$\frac{\vdash \Gamma^\perp, S'\vec{t}, P \quad \vdash \overline{B}S'\vec{x}, S'^\perp\vec{x}}{\vdash \Gamma^\perp, \nu\overline{B}\vec{t}, P} \; \nu$$

In other words, by internalizing syntactically the duality between least and greatest fixed points that exists in complemented lattices, we have also obtained the identification of induction and coinduction principles.

Example 2.5. As expected from the intended meaning of μ and ν, $\nu(\lambda p.p)$ is provable (take any provable formula as the coinvariant) and its dual $\mu(\lambda p.p)$ is not provable. More precisely, $\mu(\lambda p.p) \multimap \mathbf{0}$ and $\nu(\lambda p.p) \multimap \top$.

2.3 Comparison with Other Extensions of MALL

The logic μMALL extends MALL with first-order structure ($\forall$, $\exists$, $=$ and $\neq$) and fixed points (μ and ν). A natural question is whether fixed points can be compared with other features that bring infinite behavior, namely exponentials and second-order quantification.

In Baelde and Miller [2007], we showed that μMALL can be encoded into full second-order linear logic (LL2), that is, MALL with exponentials and second-order quantifiers, by using the well-known second-order encoding.

$$[\mu B\vec{t}] \equiv \forall S.\; !(\forall x.\; [B]S\vec{x} \multimap S\vec{x}) \multimap S\vec{t}$$

This translation highlights the fact that fixed points combine second-order aspects (the introduction of an arbitrary (co)invariant) and exponentials (the iterative behavior of the ν rule in cut elimination). The corresponding translation of μMALL derivations into LL2 is very natural; anticipating the presentation of cut elimination for μMALL, cut reductions in the original and encoded derivations should even correspond quite closely. We also provided a translation from LL2 proofs of encodings to μMALL proofs,

under natural constraints on second-order instantiations; interestingly, focusing is used to ease this reverse translation.

It is also possible to encode exponentials using fixed points, as follows.

$$[?P] \equiv \mu(\lambda p. \ \bot \oplus (p \mathbin{\invamp} p) \oplus [P]) \qquad [!P] \equiv [?P^\bot]^\bot$$

This translation trivially allows to simulate the rules of weakening (W), contraction (C) and dereliction (D) for $[?P]$ in μMALL: each one is obtained by applying the μ rule and choosing the corresponding additive disjunct. Then, the promotion rule can be obtained for the dual of the encoding. Let Γ be a sequent containing only formulas of the form $[?Q]$, and $\Gamma^\bot$ denote the tensor of the duals of those formulas, we derive $\vdash \Gamma, [!P]$ from $\vdash \Gamma, [P]$ using $\Gamma^\bot$ as a coinvariant for $[!P]$.

$$
\cfrac{
  \cfrac{
    \overline{\vdash \Gamma, \Gamma^\bot} \otimes, init
  }{}
  \quad
  \cfrac{
    \cfrac{
      \cfrac{\overline{\vdash \mathbf{1}}}{\vdash \Gamma, \mathbf{1}} \, W
    }{}
    \quad
    \cfrac{
      \cfrac{\overline{\vdash \Gamma, \Gamma^\bot} \otimes, init \quad \overline{\vdash \Gamma, \Gamma^\bot} \otimes, init}{\vdash \Gamma, \Gamma, \Gamma^\bot \otimes \Gamma^\bot}
    }{\vdash \Gamma, \Gamma^\bot \otimes \Gamma^\bot} \, C
    \quad
    \vdash \Gamma, [P]
  }{\vdash \Gamma, \mathbf{1} \mathbin{\&} (\Gamma^\bot \otimes \Gamma^\bot) \mathbin{\&} [P]}
}{\vdash \Gamma, \nu(\lambda p. \ \mathbf{1} \mathbin{\&} (p \otimes p) \mathbin{\&} [P])} \, \nu
$$

Those constructions imply that the encoding of provable statements involving exponentials is also provable in μMALL. But the converse is more problematic: not all derivations of the encoding can be translated into a derivation using exponentials. Indeed, the encoding of $[!P]$ is an infinite tree of $[P]$, and there is nothing that prevents it from containing different proofs of $[P]$, while $!P$ must be uniform, always providing the same proof of P. Finally, accordingly with these different meanings, cut reductions are different in the two systems.

It seems unlikely that second-order quantification can be encoded in μMALL, or that fixed points could be encoded using only second-order quantifiers or only exponentials. In any case, if such encodings existed they would certainly be as shallow as the encoding of exponentials, that is, at the level of provability, and not reveal a connection at the level of proofs and cut elimination like the encoding of fixed points in LL2.

2.4 Basic Metatheory

Definition 2.6. If θ is a term substitution, and Π a derivation of $\Sigma; \vdash \Gamma$, then we define $\Pi\theta$, a derivation of $\Sigma\theta; \vdash \Gamma\theta$: $\Pi\theta$ always starts with the same rule as Π, its premises being obtained naturally by applying θ to the premises of Π.

The only non-trivial case is the $\neq$ rule. Assuming that we have a derivation Π where $u \neq v$ is principal, with a subderivation Π_σ for each $\sigma \in csu(u \doteq v)$, we build a subderivation of $\Pi\theta$ for each $\sigma' \in csu(u\theta \doteq v\theta)$. Since $\theta\sigma'$ is a unifier for $u \doteq v$, it can be written as $\sigma\theta'$ for some $\sigma \in csu(u \doteq v)$. Hence, $\Pi_\sigma\theta'$ is a suitable derivation for σ'. Note that some Π_σ might be unused in that process, if σ is incompatible with θ, while others might be used infinitely many times.[3]

Note that the previous definition encompasses common signature manipulations such as permutation and extension, since it is possible for a substitution to only perform a renaming, or to translate a signature to an extended one.

---

[3]Starting with a $\neq$ rule on $x \neq y \, z$, which admits the most general unifier $[(y \, z)/x]$, and applying the substitution $\theta = [u \, v/x]$, we obtain $u \, v \neq y \, z$ which has no finite *csu*. In such a case, the infinitely many subderivations of $\Pi\theta$ would be instances of the only subderivation of Π.

We now define functoriality, a proof construction that is used to derive the following rule.

$$\frac{\vec{x};\vdash P\vec{x}, Q\vec{x}}{\Sigma;\vdash BP, \overline{B}Q} \; B$$

In functional programming terms, it corresponds to a *map* function: its type is $(Q \multimap P) \multimap (BQ \multimap BP)$ (taking $Q^{\perp}$ as Q in the above inference). Functoriality is particularly useful for dealing with fixed points: it is how we propagate reasoning/computation underneath B [Matthes 1999].

Definition 2.7 *Functoriality,* $F_B(\Pi)$. Let Π be a proof of $\vec{x};\vdash P\vec{x}, Q\vec{x}$ and B be a monotonic operator such that $\Sigma \vdash B : (\vec{\gamma} \to o) \to o$. We define $F_B(\Pi)$, a derivation of $\Sigma;\vdash BP, \overline{B}Q$, by induction on the maximum depth of occurrences of p in Bp:

— When $B = \lambda p.\ P'$, $F_B(\Pi)$ is an instance of *init* on P'.
— When $B = \lambda p.\ p\vec{t}$, $F_B(\Pi)$ is $\Pi[\vec{t}/\vec{x}]$.
— Otherwise, we perform an η-expansion based on the toplevel connective of B and conclude by induction hypothesis. We only show half of the connectives, because dual connectives are treated symmetrically. There is no case for units, equality, and disequality since they are treated as part of the vacuous abstraction case.
   When $B = \lambda p.\ B_1 p \otimes B_2 p$:

$$\frac{\dfrac{F_{B_1}(\Pi)}{\Sigma;\vdash B_1 P, \overline{B}_1 Q} \quad \dfrac{F_{B_2}(\Pi)}{\Sigma;\vdash B_2 P, \overline{B}_2 Q}}{\dfrac{\Sigma;\vdash B_1 P \otimes B_2 P, \overline{B}_1 Q, \overline{B}_2 Q}{\Sigma;\vdash B_1 P \otimes B_2 P, \overline{B}_1 Q \,\invamp\, \overline{B}_2 Q}\invamp} \otimes \quad .$$

When $B = \lambda p.\ B_1 p \oplus B_2 p$:

$$\frac{\dfrac{\dfrac{F_{B_1}(\Pi)}{\Sigma;\vdash B_1 P, \overline{B}_1 Q}}{\Sigma;\vdash B_1 P \oplus B_2 P, \overline{B}_1 Q}\oplus \quad \dfrac{\dfrac{F_{B_2}(\Pi)}{\Sigma;\vdash B_2 P, \overline{B}_2 Q}}{\Sigma;\vdash B_1 P \oplus B_2 P, \overline{B}_2 Q}\oplus}{\Sigma;\vdash B_1 P \oplus B_2 P, \overline{B}_1 Q \,\&\, \overline{B}_2 Q} \& \quad .$$

When $B = \lambda p.\ \exists x.\ B'px$:

$$\frac{\dfrac{\dfrac{F_{B'\bullet x}(\Pi)}{\Sigma, x;\vdash B'Px, \overline{B}'Qx}}{\Sigma, x;\vdash \exists x.\ B'Px, \overline{B}'Qx}\exists}{\Sigma;\vdash \exists x.\ B'Px, \forall x.\ \overline{B}'Qx} \forall \quad .$$

When $B = \lambda p.\ \mu(B'p)\vec{t}$, we show that $\nu(\overline{B}'P^{\perp})$ is a coinvariant of $\nu(\overline{B}'Q)$:

$$\frac{\dfrac{}{\Sigma;\vdash \mu(B'P)\vec{t}, \nu(\overline{B}'P^{\perp})\vec{t}}init \quad \dfrac{\dfrac{F_{(\lambda p.B'p(\mu(B'P))\vec{x})}(\Pi)}{\vec{x};\vdash (B'P)(\mu(B'P))\vec{x}, (\overline{B}'Q)(\nu(\overline{B}'P^{\perp}))\vec{x}}}{\vec{x};\vdash \mu(B'P)\vec{x}, (\overline{B}'Q)(\nu(\overline{B}'P^{\perp}))\vec{x}}\mu}{\Sigma;\vdash \mu(B'P)\vec{t}, \nu(\overline{B}'Q)\vec{t}} \nu \quad .$$

PROPOSITION 2.8 ATOMIC INITIAL RULE. *We call* atomic *the init rules acting on atoms or fixed points. The general rule init is derivable from atomic initial rules.*

PROOF. By induction on P, we build a derivation of $\vdash P^{\perp}, P$ using only atomic axioms. If P is not an atom or a fixed point expression, we perform an η-expansion as in the previous definition and conclude by induction hypothesis. Note that although the identity on fixed points can be expanded, it can never be eliminated: repeated expansions do not terminate in general. □

The constructions used above can be used to establish the *canonicity* of all our logical connectives: if a connective is duplicated into, say, red and blue variants equipped with the same logical rules, then those two versions are equivalent. Intuitively, it means that our connectives define a unique logical concept. This is a known property of the connectives of first-order MALL, we show it for μ and its copy $\hat{\mu}$ by using our color-blind expansion.

$$\cfrac{\cfrac{}{\vdash \hat{\nu}\overline{B}\,\vec{t}, \hat{\mu}B\vec{t}}\; init \qquad \cfrac{\cfrac{}{\vdash \overline{B}(\hat{\nu}\overline{B})\vec{x}, B(\hat{\mu}B)\vec{x}}\; init}{\vdash \overline{B}(\hat{\nu}\overline{B})\vec{x}, \hat{\mu}B\vec{x}}\; \hat{\mu}}{\vdash \nu\overline{B}\,\vec{t}, \hat{\mu}B\vec{t}}\; \nu$$

PROPOSITION 2.9. *The following inference rule is derivable:*

$$\cfrac{\vdash \Gamma, B(\nu B)\vec{t}}{\vdash \Gamma, \nu B\vec{t}}\; \nu R \;.$$

PROOF. The unfolding νR is derivable from ν, using $B(\nu B)$ as the coinvariant S. The proof of coinvariance $\vdash B(B(\nu B))\vec{x}, \overline{B}(\mu\overline{B})\vec{x}$ is obtained by functoriality on $\vdash B(\nu B)\vec{x}, \mu\overline{B}\vec{x}$, itself obtained from μ and *init*. □

Example 2.10. In general the least fixed point entails the greatest. The following is a proof of $\mu B\vec{t} \multimap \nu B\vec{t}$, showing that μB is a coinvariant of νB.

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

The greatest fixed point entails the least fixed point when the fixed points are *Noetherian*, that is, predicate operators have vacuous second-order abstractions. Finally, the νR rule allows to derive $\mu B\vec{t} \circ\!\!-\!\!\circ B(\mu B)\vec{t}$, or equivalently $\nu B\vec{t} \circ\!\!-\!\!\circ B(\nu B)\vec{t}$.

2.5 Polarities of Connectives

It is common to classify inference rules between invertible and noninvertible ones. In linear logic, we can use the refined notions of *positivity* and *negativity*. A formula P is said to be positive (resp. Q is said to be negative) when $P \circ\!\!-\!\!\circ \;!P$ (resp. $Q \circ\!\!-\!\!\circ \;?Q$). A logical connective is said to be positive (resp. negative) when it preserves positivity (resp. negativity). For example, $\otimes$ is positive since $P \otimes P'$ is positive whenever P and P' are. This notion is more semantical than invertibility, and has the advantage of actually saying something about noninvertible connectives/rules. Although it does not seem at first sight to be related to proof-search, positivity turns out to play an important role in the understanding and design of focused systems [Danos et al. 1993, 1995; Laurent 2002; Laurent et al. 2005; Liang and Miller 2007].

Since μMALL does not have exponentials, it is not possible to talk about positivity following this traditional definition. Instead, we are going to take a backwards

approach: we shall first define which connectives are negative, and then check that the obtained negative formulas have a property close to the original negativity. This does not trivialize the question at all: it turns out that only one classification allows to derive the expected property. We refer the interested reader to Baelde [2008a] for the extension of that proof to μLL, that is, μMALL with exponentials, where we follow the traditional approach.

Definition 2.11. We classify as *negative* the following connectives: $\mathbin{\rotatebox[origin=c]{180}{\&}}$, $\bot$, &, $\top$, $\forall$, $\neq$, ν. Their duals are called *positive*. A formula is said to be negative (resp. positive) when all of its connectives are negative (resp. positive). Finally, an operator $\lambda p \lambda \vec{x}.Bp\vec{x}$ is said to be negative (resp. positive) when the formula $Bp\vec{x}$ is negative (resp. positive).

Notice, for example, that $\lambda p \lambda \vec{x}.p\vec{x}$ is both positive and negative. But $\mu p.p$ is only positive while $\nu p.p$ is only negative. Atoms (and formulas containing atoms) are neither negative nor positive: indeed, they offer no structure[4] from which the following fundamental property could be derived.

PROPOSITION 2.12. *The following structural rules are admissible for any negative formula P.*

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

We can already note that this proposition could not hold if μ was negative, since $\mu(\lambda p.p)$ cannot be weakened (there is obviously no cut-free proof of $\vdash \mu(\lambda p.p), \mathbf{1}$).

PROOF. We first prove the admissibility of W. This rule can be obtained by cutting a derivation of $\Sigma; \vdash P, \mathbf{1}$. We show more generally that for any collection of negative formulas $(P_i)_i$, there is a derivation of $\vdash (P_i)_i, \mathbf{1}$. This is done by induction on the total size of $(P_i)_i$, counting one for each connective, unit, atom or predicate variable but ignoring terms. The proof is trivial if the collection is empty. Otherwise, if P_0 is a disequality we conclude by induction with one less formula, and the size of the others unaffected by the first-order instantiation; if it is $\top$ our proof is done; if it is $\bot$ then P_0 disappears and we conclude by induction hypothesis. The $\mathbin{\rotatebox[origin=c]{180}{\&}}$ case is done by induction hypothesis, the resulting collection has one more formula but is smaller; the & makes use of two instances of the induction hypothesis; the $\forall$ case makes use of the induction hypothesis with an extended signature but a smaller formula. Finally, the ν case is done by applying the ν rule with $\bot$ as the invariant.

$$\frac{\dfrac{\vdash (P_i)_i, \mathbf{1}}{\vdash \bot, (P_i)_i, \mathbf{1}} \quad \vdash B(\lambda \vec{x}.\bot)\vec{x}, \mathbf{1}}{\vdash \nu B\vec{t}, (P_i)_i, \mathbf{1}}$$

The two subderivations are obtained by induction hypothesis. For the second one there is only one formula, namely $B(\lambda \vec{x}.\bot)\vec{x}$, which is indeed negative (by monotonicity of B) and smaller than νB.

We also derive contraction (C) using a cut, this time against a derivation of $\vdash (P \mathbin{\rotatebox[origin=c]{180}{\&}} P)^\perp, P$. A generalization is needed for the greatest fixed point case, and we derive the following for any negative n-ary operator A:

$$\vdash (A(\nu B_1)\dots(\nu B_n) \mathbin{\rotatebox[origin=c]{180}{\&}} A(\nu B_1)\dots(\nu B_n))^\perp, A(\nu B_1 \mathbin{\rotatebox[origin=c]{180}{\&}} \nu B_1)\dots(\nu B_n \mathbin{\rotatebox[origin=c]{180}{\&}} \nu B_n).$$

----

[4]This essential aspect of atoms makes them often less interesting or even undesirable. For example, in our work on minimal generic quantification [Baelde 2008b] we show and exploit the fact that this third quantifier can be defined in μLJ *without atoms*.

We prove this by induction on A as follows.

— It is trivial if A is a disequality, $\top$ or $\bot$.
— If A is a projection $\lambda\vec{p}.\ p_i\vec{t}$, we have to derive $\vdash (\nu B_i\vec{t} \,\mathbin{\bindnasrepma}\, \nu B_i\vec{t})^\bot, \nu B_i\vec{t} \,\mathbin{\bindnasrepma}\, \nu B_i\vec{t}$, which is an instance of *init*.
— If A is $\lambda\vec{p}.\ A_1\vec{p} \,\mathbin{\bindnasrepma}\, A_2\vec{p}$, we can combine our two induction hypotheses to derive the following:

$$\vdash ((A_1(\nu B_i)_i \,\mathbin{\bindnasrepma}\, A_1(\nu B_i)_i) \,\mathbin{\bindnasrepma}\, (A_2(\nu B_i)_i \,\mathbin{\bindnasrepma}\, A_2(\nu B_i)_i))^\bot, A_1(\nu B_i)_i \,\mathbin{\bindnasrepma}\, A_2(\nu B_i)_i.$$

We conclude by associativity-commutativity of the tensor, which amounts to use cut against an easily obtained derivation of $\vdash ((P_1\mathbin{\bindnasrepma}P_2)\mathbin{\bindnasrepma}(P_1\mathbin{\bindnasrepma}P_2)), ((P_1\mathbin{\bindnasrepma}P_1)\mathbin{\bindnasrepma}(P_2\mathbin{\bindnasrepma}P_2))^\bot$ for $P_j := A_j(\nu B_i)_i$.
— If A is $\lambda\vec{p}.\ A_1\vec{p} \mathbin{\&} A_2\vec{p}$ we introduce the additive conjunction and have to derive two similar premises:

$$\vdash ((A_1 \mathbin{\&} A_2)(\nu B_i)_i \,\mathbin{\bindnasrepma}\, (A_1 \mathbin{\&} A_2)(\nu B_i)_i)^\bot, A_j(\nu B_i \,\mathbin{\bindnasrepma}\, \nu B_i)_i \ \text{ for } j \in \{1, 2\}$$

To conclude by induction hypothesis, we have to choose the correct projections for the negated $\&$. Since the $\&$ is under the $\mathbin{\bindnasrepma}$, we have to use a cut; one can derive in general $\vdash ((P_1 \mathbin{\&} P_2) \,\mathbin{\bindnasrepma}\, (P_1 \mathbin{\&} P_2))^\bot, P_j \,\mathbin{\bindnasrepma}\, P_j$ for $j \in \{1, 2\}$.
— When A is $\lambda\vec{p}.\ \forall x.\ A'\vec{p}x$, the same scheme applies: we introduce the universal variable and instantiate the two existential quantifiers under the $\mathbin{\bindnasrepma}$ thanks to a cut.
— Finally, we treat the greatest fixed point case: A is $\lambda\vec{p}.\ \nu(A'\vec{p})\vec{t}$. Let B_{n+1} be $A'(\nu B_i)_{i\leq n}$. We have to build a derivation of

$$\vdash (\nu B_{n+1}\vec{t} \,\mathbin{\bindnasrepma}\, \nu B_{n+1}\vec{t})^\bot, \nu(A'(\nu B_i \,\mathbin{\bindnasrepma}\, \nu B_i)_i)\vec{t}.$$

We use the ν rule, showing that $\nu B_{n+1} \,\mathbin{\bindnasrepma}\, \nu B_{n+1}$ is a coinvariant of $\nu(A'(\nu B_i \,\mathbin{\bindnasrepma}\, \nu B_i)_i)$. The left subderivation of the ν rule is thus an instance of *init*, and the coinvariance derivation is as follows.

$$\frac{\vdash (A'(\nu B_i)_i(\nu B_{n+1})\vec{x} \,\mathbin{\bindnasrepma}\, A'(\nu B_i)_i(\nu B_{n+1})\vec{x})^\bot, A'(\nu B_i \,\mathbin{\bindnasrepma}\, \nu B_i)_i(\nu B_{n+1} \,\mathbin{\bindnasrepma}\, \nu B_{n+1})\vec{x} \quad \Pi'}{\vdash (\nu B_{n+1}\vec{x} \,\mathbin{\bindnasrepma}\, \nu B_{n+1}\vec{x})^\bot, A'(\nu B_i \,\mathbin{\bindnasrepma}\, \nu B_i)_i(\nu B_{n+1} \,\mathbin{\bindnasrepma}\, \nu B_{n+1})\vec{x}} \ cut$$

Here, Π' derives $\vdash (\nu B_{n+1}\vec{x} \,\mathbin{\bindnasrepma}\, \nu B_{n+1}\vec{x})^\bot, A'(\nu B_i)_i(\nu B_{n+1})\vec{x} \,\mathbin{\bindnasrepma}\, A'(\nu B_i)_i(\nu B_{n+1})\vec{x}$, unfolding νB_{n+1} under the tensor. We complete our derivation by induction hypothesis, with the smaller operator expression A' and B_{n+1} added to the $(B_i)_i$.   □

The previous property yields some interesting remarks about the expressiveness of μMALL. It is easy to see that provability is undecidable in μMALL, by encoding (terminating) executions of a Turing machine as a least fixed point. But this kind of observation does not say anything about what theorems can be derived, that is, the complexity of reasoning/computation allowed in μMALL. Here, the negative structural rules derived in Proposition 2.12 come into play. Although our logic is linear, it enjoys those derived structural rules for a rich class of formulas: for example, *nat* is positive, hence reasoning about natural numbers allows contraction and weakening, just like in an intuitionistic setting. Although the precise complexity of the normalization of μMALL is unknown, we have adapted some remarks from [Alves et al. 2006; Burroni 1986; Girard 1987] to build an encoding of primitive recursive functions in μMALL [Baelde 2008a]; in other words, all primitive recursive functions can be proved total in μMALL.

2.6 Examples

We shall now give a few theorems derivable in μMALL. Although we do not provide their derivations here but only brief descriptions of how to obtain them, we stress that all of these examples are proved naturally. The reader will note that although μMALL is linear, these derivations are intuitive and their structure resembles that of proofs in intuitionistic logic. We also invite the reader to check that the μ-focusing system presented in Section 4 is a useful guide when deriving these examples, leaving only the important choices. It should be noted that atoms are not used in this section; in fact, atoms are rarely useful in μMALL, as its main application is to reason about (fully defined) fixed points.

Following the definition of *nat* from Example 2.4, we define a few least fixed points expressing basic properties of natural numbers. Note that all these definitions are positive.

$$even \overset{def}{=} \mu(\lambda E \lambda x.\ x = 0 \oplus \exists y.\ x = s\,(s\,y) \otimes E\ y)$$

$$plus \overset{def}{=} \mu(\lambda P \lambda a \lambda b\,\lambda c.\ a = 0 \otimes b = c$$
$$\oplus\ \exists a' \exists c'.a = s\,a' \otimes c = s\,c' \otimes P\ a'\ b\ c')$$

$$leq \overset{def}{=} \mu(\lambda L \lambda x \lambda y.\ x = y \oplus \exists y'.\ y = s\,y' \otimes L\ x\ y')$$

$$half \overset{def}{=} \mu(\lambda H \lambda x \lambda h.\ (x = 0 \oplus x = s\,0) \otimes h = 0$$
$$\oplus\ \exists x' \exists h'.\ x = s\,(s\,x') \otimes h = s\,h' \otimes H\ x'\ h')$$

$$ack \overset{def}{=} \mu(\lambda A \lambda m \lambda n \lambda a.\ m = 0 \otimes a = s\,n$$
$$\oplus\ (\exists p.\ m = s\,p \otimes n = 0 \otimes A\ p\ (s\,0)\ a)$$
$$\oplus\ (\exists p \exists q \exists b.\ m = s\,p \otimes n = s\,q \otimes A\ m\ q\ b \otimes A\ p\ b\ a))$$

The following statements are theorems in μMALL. 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.

$$\vdash \forall x.\ nat\ x \multimap even\ x \oplus even\,(s\,x)$$
$$\vdash \forall x.\ nat\ x \multimap \forall y \exists z.\ plus\ x\ y\ z$$
$$\vdash \forall x.\ nat\ x \multimap plus\ x\ 0\ x$$
$$\vdash \forall x.\ nat\ x \multimap \forall y.\ nat\ y \multimap \forall z.\ plus\ x\ y\ z \multimap nat\ z$$

In the last theorem, the assumption (*nat x*) is not needed and can be weakened, thanks to Proposition 2.12. In order to prove ($\forall x.\ nat\ x \multimap \exists h.\ half\ x\ h$) the context does not provide an invariant that is strong enough. A typical solution is to use complete induction, that is, use the strengthened invariant ($\lambda x.\ nat\ x \otimes \forall y.\ leq\ y\ x \multimap \exists h.\ half\ y\ h$).

We do not know of any proof of totality for a nonprimitive recursive function in μMALL. In particular, we have no proof of $\forall x \forall y.\ nat\ x \multimap nat\ y \multimap \exists z.\ ack\ x\ y\ z$. The corresponding intuitionistic theorem can be proved using nested inductions, but it does not lead to a linear proof since it requires to contract an implication hypothesis (in μMALL, the dual of an implication is a tensor, which is not negative and thus cannot *a priori* be contracted).

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

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

Reflexivity of simulation ($\forall p.\ sim\ p\ p$) is proved easily by coinduction 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 coinduction on ($sim\ p\ r$) with the coinvariant ($\lambda p \lambda r.\ \exists q.\ sim\ p\ q \otimes sim\ q\ r$). 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 instances of *step*. Notice that these two examples are also cases where the context gives a coinvariant.

3. NORMALIZATION

In Baelde and Miller [2007], we provided an indirect proof of normalization based on the second-order encoding of μMALL. However, that proof relied on the normalization of second-order linear logic extended with first-order quantifiers, and more importantly equality, but this extension of Girard's result for propositional second-order linear logic is only a (mild) conjecture. Moreover, such an indirect proof does not provide cut reduction rules, which usually illuminate the structure and meaning of a logic. In this article, we give the first direct and full proof of normalization for μMALL: we provide a system of reduction rules for eliminating cuts, and show that it is weakly normalizing by using the candidates of reducibility technique [Girard 1987]. Establishing strong normalization would be useful, but we leave it to further work. Note that the candidates of reducibility technique is quite modular in that respect: in fact, Girard [1987] provided only a proof of weak normalizability together with a conjectured standardization lemma from which strong normalization would follow. Also note, by the way, that Girard's proof applies to proof nets, while we shall work directly within sequent calculus; again, the adaptation is quite simple. Finally, the candidate of reducibility is also modular in that it relies on a compositional interpretation of connectives, so that our normalization proof (unlike the earlier one) should extend easily to exponentials and second-order quantification using their usual interpretations.

Our proof can be related to similar work in other settings. While it would technically have been possible to interpret fixed points as candidates through their second-order encoding, we found it more appealing to directly interpret them as fixed point candidates. In that respect, our work can be seen as an adaptation of the ideas from Mendler [1991] and Matthes [1999] to the classical linear setting, where candidates of reducibility are more naturally expressed as biorthogonals. This adaptation turns out to work really well, and the interpretation of least fixed points as least fixed points on candidates yields a rather natural proof, notably proceeding by metalevel induction on that fixed point construction. Also related, of course, is the work on definitions; although we consider a linear setting and definitions have been studied in intuitionistic logic, we believe that our proof could be adapted, and contributes to the understanding of similar notions. In addition to the limitations of definitions over fixed points, the only published proof of cut elimination [Momigliano and Tiu 2003; Tiu 2004] further restricts definitions to strictly positive ones, and limits the coinduction rule to coinvariants of smaller "level" than the considered coinductive object. However, those two restrictions have been removed in Tiu and Momigliano [2010], which relies (like our proof) on a full candidate of reducibility argument rather than the earlier nonparametrized reducibility, and essentially follows (unlike our proof) a second-order encoding of definitions.

We now proceed with the proof, defining cut reductions and then showing their normalization. Instead of writing proof trees, we shall often use an informal term notation for proofs, when missing details can be inferred from the context. We notably write $cut(\Pi; \Pi')$ for a cut, and more generally $cut(\Pi; \vec{\Pi}')$ for the sequence of cuts $cut(\ldots cut(\Pi; \Pi'_1) \ldots; \Pi'_n)$. We also use notations such as $\Pi \otimes \Pi'$, $\mu\Pi$, $\nu(\Pi, \Theta)$, etc. Although the first-order structure does not play a role in the termination and complexity of reductions, we decided to treat it directly in the proof, rather than evacuating it in a first step. We tried to keep it readable, but encourage the reader to translate the most technical parts for the purely propositional case in order to extract their core.

3.1 Reduction Rules

Rules reduce instances of the cut rule, and are separated into auxiliary and main rules. Most of the rules are the same as for MALL. For readability, we do not show the signatures Σ when they are not modified by reductions, leaving to the reader the simple task of inferring them.

3.1.1 Auxiliary Cases. If a subderivation does not start with a logical rule in which the cut formula is principal, its first rule is permuted with the cut. We only present the commutations for the left subderivation, the situation being perfectly symmetric.

— If the subderivation starts with a cut, splitting Γ into Γ', Γ'', we reduce as follows.

$$
\cfrac{\cfrac{\vdash \Gamma', P^\perp, Q^\perp \quad \vdash \Gamma'', Q}{\vdash \Gamma', \Gamma'', P^\perp} \; cut \quad \vdash P, \Delta}{\vdash \Gamma', \Gamma'', \Delta} \; cut
$$

$$\downarrow$$

$$
\cfrac{\cfrac{\vdash \Gamma', Q^\perp, P^\perp \quad \vdash P, \Delta}{\vdash \Gamma', \Delta, Q^\perp} \; cut \quad \vdash Q, \Gamma''}{\vdash \Gamma', \Gamma'', \Delta} \; cut
$$

Note that this reduction alone leads to cycles, hence our system is trivially not strongly normalizing. This is only a minor issue, which could be solved, for example, by using proof nets or a classical multicut rule (which amounts to incorporate the required amount of proof net flexibility into sequent calculus).

— Identity between a cut formula and a formula from the conclusion: Γ is restricted to the formula P and the left subderivation is an axiom. The cut is deleted and the right subderivation is now directly connected to the conclusion instead of the cut formula.

$$
\cfrac{\cfrac{}{\vdash P, P^\perp} \; init \quad \cfrac{\Pi}{\vdash P, \Delta}}{\vdash P, \Delta} \; cut \quad \longrightarrow \quad \cfrac{\Pi}{\vdash P, \Delta}
$$

— When permuting a cut and a $\otimes$, the cut is dispatched according to the splitting of the cut formula. When permuting a cut and a &, the cut is duplicated. The rules $\invamp$ and $\oplus$ are easily commuted down the cut.

— The commutations of $\top$ and $\perp$ are simple, and there is none for $\mathbf{1}$ nor $\mathbf{0}$.

— When $\forall$ is introduced, it is permuted down and the signature of the other derivation is extended. The $\exists$ rule is permuted down without any problem.

— There is no commutation for equality (=). When a disequality ($\neq$) is permuted down, the other premise is duplicated and instantiated as follows.

$$\cfrac{\left\{\cfrac{\Pi_\theta}{\Sigma\theta;\vdash \Gamma'\theta, P^\perp\theta}\right\}}{\cfrac{\Sigma;\vdash \Gamma', u \neq v, P^\perp}{\Sigma;\vdash \Gamma', u \neq v, \Delta}\ cut} \neq \quad \cfrac{\Pi'}{\Sigma;\vdash P, \Delta}$$

$$\downarrow$$

$$\cfrac{\left\{\cfrac{\cfrac{\Pi_\theta}{\Sigma\theta;\vdash \Gamma'\theta, P^\perp\theta}\quad \cfrac{\Pi'\theta}{\Sigma\theta;\vdash P\theta, \Delta\theta}}{\Sigma\theta;\vdash \Gamma'\theta, \Delta\theta}\ cut\right\}}{\Sigma;\vdash \Gamma', u \neq v, \Delta}\neq$$

— $\Gamma = \Gamma', \mu B\vec{t}$ and that least fixed point is introduced.

$$\cfrac{\cfrac{\vdash \Gamma', B(\mu B)\vec{t}, P^\perp}{\vdash \Gamma', \mu B\vec{t}, P^\perp}\ \mu \quad \vdash P, \Delta}{\vdash \Gamma', \mu B\vec{t}, \Delta}\ cut \quad \longrightarrow \quad \cfrac{\cfrac{\vdash \Gamma', B(\mu B)\vec{t}, P^\perp \quad \vdash P, \Delta}{\vdash \Gamma', B(\mu B)\vec{t}, \Delta}\ cut}{\vdash \Gamma', \mu B\vec{t}, \Delta}\ \mu$$

— $\Gamma = \Gamma', \nu B\vec{t}$ and that greatest fixed point is introduced.

$$\cfrac{\cfrac{\vdash \Gamma', S\vec{t}, P^\perp \quad \vdash S\vec{x}^\perp, BS\vec{x}}{\vdash \Gamma', \nu B\vec{t}, P^\perp}\ \nu \quad \vdash P, \Delta}{\vdash \Gamma', \nu B\vec{t}, \Delta}\ cut$$

$$\downarrow$$

$$\cfrac{\cfrac{\vdash \Gamma', S\vec{t}, P^\perp \quad \vdash P, \Delta}{\vdash \Gamma', S\vec{t}, \Delta}\ cut \quad \vdash S\vec{x}^\perp, BS\vec{x}}{\vdash \Gamma', \nu B\vec{t}, \Delta}\ \nu$$

3.1.2 Main Cases. When a logical rule is applied on the cut formula on both sides, one of the following reductions applies.

— In the multiplicative case, Γ is split into (Γ', Γ'') and we cut the subformulas.

$$\cfrac{\cfrac{\vdash \Gamma', P' \quad \vdash \Gamma'', P''}{\vdash \Gamma', \Gamma'', P' \otimes P''}\ \otimes \quad \cfrac{\vdash P'^\perp, P''^\perp, \Delta}{\vdash P'^\perp \mathbin{\rotatebox[origin=c]{180}{\&}} P''^\perp, \Delta}\ \mathbin{\rotatebox[origin=c]{180}{\&}}}{\vdash \Gamma', \Gamma'', \Delta}\ cut$$

$$\downarrow$$

$$\cfrac{\vdash \Gamma', P' \quad \cfrac{\vdash \Gamma'', P'' \quad \vdash P'^\perp, P''^\perp, \Delta}{\vdash P'^\perp, \Gamma'', \Delta}\ cut}{\vdash \Gamma', \Gamma'', \Delta}\ cut$$

— In the additive case, we select the appropriate premise of &.

$$\cfrac{\cfrac{\vdash \Gamma, P_i}{\vdash \Gamma, P_0 \oplus P_1}\ \oplus \quad \cfrac{\vdash \Delta, P_0^\perp \quad \vdash \Delta, P_1^\perp}{\vdash \Delta, P_0^\perp \& P_1^\perp}\ \&}{\vdash \Gamma, \Delta}\ cut \quad \longrightarrow \quad \cfrac{\vdash \Gamma, P_i \quad \vdash \Delta, P_i^\perp}{\vdash \Gamma, \Delta}\ cut$$

— The $\mathbf{1}/\bot$ case reduces to the subderivation of $\bot$. There is no case for $\top/\mathbf{0}$.

— In the first-order quantification case, we perform a proof instantiation.

$$\cfrac{\cfrac{\Pi_l}{\Sigma;\vdash \Gamma, Pt^\bot} \quad \cfrac{\cfrac{\Pi_r}{\Sigma, x;\vdash Px, \Delta}}{\Sigma;\vdash \forall x.\ Px, \Delta}\ \forall}{\Sigma;\vdash \Gamma, \Delta}\ cut \quad\longrightarrow\quad \cfrac{\cfrac{\Pi_l}{\Sigma;\vdash \Gamma, Pt^\bot} \quad \cfrac{\Pi_r[t/x]}{\Sigma;\vdash Pt, \Delta}}{\Sigma;\vdash \Gamma, \Delta}\ cut$$

where $\Sigma;\vdash \Gamma, \exists x.\ Px^\bot$ is derived from Π_l by $\exists$.

— The equality case is trivial, the interesting part concerning this connective lies in the proof instantiations triggered by other reductions. Since we are considering two terms that are already equal, we have $csu(u \doteq u) = \{id\}$ and we can simply reduce to the subderivation corresponding to the identity substitution.

$$\cfrac{\cfrac{}{\Sigma;\vdash u = u}\ =\quad \cfrac{\cfrac{\Pi_{id}}{\Sigma;\vdash \Delta}}{\Sigma;\vdash u \neq u, \Delta}\ \neq}{\Sigma;\vdash \Delta}\ cut \quad\longrightarrow\quad \cfrac{\Pi_{id}}{\Sigma;\vdash \Delta}$$

— Finally in the fixed point case, we make use of the functoriality transformation for propagating the coinduction/recursion under B.

$$\cfrac{\cfrac{\Pi'_l}{\Sigma;\vdash \Gamma, B(\mu B)\vec{t}}\ \mu \quad \cfrac{\cfrac{\Pi'_r}{\Sigma;\vdash \Delta, S\vec{t}} \quad \cfrac{\Theta}{\vec{x};\vdash S\vec{x}^\bot, \overline{B}S\vec{x}}}{\Sigma;\vdash \Delta, \nu \overline{B}\vec{t}}\ \nu}{\Sigma;\vdash \Gamma, \Delta}\ cut$$

$$\downarrow$$

$$\cfrac{\cfrac{\Pi'_r}{\Sigma;\vdash \Delta, S\vec{t}} \quad \cfrac{\cfrac{\Theta[\vec{t}/\vec{x}]}{\Sigma;\vdash S\vec{t}^\bot, \overline{B}S\vec{t}} \quad \cfrac{\cfrac{F_{B\bullet\vec{t}}(\nu(Id, \Theta))}{\Sigma;\vdash BS^\bot\vec{t}, \overline{B}(\nu\overline{B})\vec{t}} \quad \cfrac{\Pi'_l}{\Sigma;\vdash B(\mu B)\vec{t}, \Gamma}}{\Sigma;\vdash BS^\bot\vec{t}, \Gamma}\ cut}{\Sigma;\vdash S\vec{t}^\bot, \Gamma}\ cut}{\Sigma;\vdash \Gamma, \Delta}\ cut$$

One-step reduction $\Pi \to \Pi'$ is defined as the congruence generated by these rules. We now seek to establish that such reductions can be applied to transform any derivation into a cut-free one. However, since we are dealing with transfinite (infinitely branching) proof objects, there are trivially derivations which cannot be reduced into a cut-free form in a finite number of steps. A possibility would be to consider transfinite reduction sequences, relying on a notion of convergence for defining limits. A simpler solution, enabled by the fact that our infinity only happens "in parallel," is to define inductively the transfinite reflexive transitive closure of one-step reduction.

Definition 3.1 *Reflexive transitive closure,* $\mathcal{WN}$. We define inductively $\Pi \to^* \Xi$ to hold when (1) $\Pi \to \Xi$, (2) $\Pi \to^* \Pi'$ and $\Pi' \to^* \Xi$, or (3) Π and Ξ start with the same rule and their premises are in relation (i.e., for some rule $\mathcal{R}$, $\Pi = \mathcal{R}(\Pi_i)_i$, $\Xi = \mathcal{R}(\Xi_i)_i$ and each $\Pi_i \to^* \Xi_i$). We say that Π *normalizes* when there exists a cut-free derivation Π' such that $\Pi \to^* \Pi'$. We denote by $\mathcal{WN}$ the set of all normalizing derivations.

From (1) and (2), it follows that if Π reduces to Ξ in $n > 0$ steps, then $\Pi \to^* \Xi$. From (3) it follows that $\Pi \to^* \Pi$ for any Π. In the finitely branching case, that is, if the $\neq$ connective was removed or the system ensured finite *csu*, the role of (3) is only

to ensure reflexivity. In the presence of infinitely branching rules, however, it also plays the important role of packaging an infinite number of reductions. In the finitely branching case, one can show that $\Pi \to^* \Xi$ implies that there is a finite reduction sequence from Π to Ξ (by induction on $\Pi \to^* \Xi$), and so our definition of normalization corresponds to the usual notion of weak normalization in that case.

PROPOSITION 3.2. *If $\Pi \to \Xi$, then $\Pi\theta \to^* \Xi\theta$.*

PROOF. By induction on Π. If the redex is not at toplevel but in an immediate subderivation Π', then the corresponding subderivations in $\Pi\theta$ shall be reduced. If the first rule of Π is disequality, there may be zero, several or infinitely many subderivations of $\Pi\theta$ of the form $\Pi'\theta'$. Otherwise there is only one such subderivation. In both cases, we show $\Pi\theta \to^* \Xi\theta$ by (3), using the induction hypothesis for the subderivations where the redex is, and reflexivity of $\to^*$ for the others.

If the redex is at toplevel, then $\Pi\theta \to \Xi\theta$. The only nontrivial cases are the two reductions involving $\neq$. In the auxiliary case, we have:

$$
\begin{array}{ccc}
cut(\neq(\Pi_\sigma)_{\sigma \in csu(u \doteq v)}; \Pi_r) & \longrightarrow & \neq(cut(\Pi_\sigma; \Pi_r\sigma))_\sigma \\
\Big\downarrow{\scriptstyle\theta} & & \Big\downarrow{\scriptstyle\theta} \\
cut(\neq(\Pi'_{\sigma'})_{\sigma' \in csu(u\theta \doteq v\theta)}; \Pi_r\theta) & \dashrightarrow & \neq(cut(\Pi'_{\sigma'}; (\Pi_r\theta)\sigma'))_{\sigma'}
\end{array}
$$

By Definition 2.6, $\Pi'_{\sigma'} = \Pi_\sigma\sigma''$ for $\theta\sigma' = \sigma\sigma''$, $\sigma \in csu(u \doteq v)$. Applying θ on the reduct of Π, we obtain for each σ' the subderivation $cut(\Pi_\sigma; \Pi_r\sigma)\sigma'' = cut(\Pi_\sigma\sigma''; \Pi_r\sigma\sigma'') = cut(\Pi'_{\sigma'}; \Pi_r\theta\sigma')$.

In the main case, $\Pi = cut(\neq(\Pi_{id}); u = u) \to \Pi_{id}$ and $\Pi\theta = cut(\neq(\Pi'_{id}); u\theta = u\theta) \to \Pi'_{id} = \Pi_{id}\theta$. □

PROPOSITION 3.3. *If Π is normalizing, then so is $\Pi\theta$.*

PROOF. Given a cut-free derivation Π' such that $\Pi \to^* \Pi'$, we show that $\Pi\theta \to^* \Pi'\theta$ by a simple induction on $\Pi \to^* \Pi'$, making use of the previous proposition. □

PROPOSITION 3.4. *We say that Ξ is an Id-simplification of Π if it is obtained from Π by reducing an arbitrary, potentially infinite number of redexes $cut(\Theta; Id)$ into Θ. If Ξ is an Id-simplification of Π and Π is normalizable, then so is Ξ.*

PROOF. We show more generally that if Ξ is a simplification of Π and $\Pi \to^* \Pi'$ then $\Xi \to^* \Xi'$ for some simplification Ξ' of Π'. This is easily done by induction on $\Pi \to^* \Pi'$, once we will have established the following fact: *If Ξ is a simplification of Π and $\Pi \to \Pi'$, then $\Xi \to^* \Xi'$ for a simplification Ξ' of Π'.* If the redex in Π does not involve simplified cuts, the same reduction can be performed in Ξ, and the result is a simplification of Π' (note that this could erase or duplicate some simplifications). If the reduction is one of the simplications then Ξ itself is a simplification of Π'. If a simplified cut is permuted with another cut (simplified or not) Ξ is also a simplification of Π'. Finally, other auxiliary reductions on a simplified cut also yield reducts of which Ξ is already a simplification (again, simplifications may be erased or duplicated). □

3.2 Reducibility Candidates

Definition 3.5 *Type*. A proof of type P is a proof with a distinguished formula P among its conclusion sequent. We denote by Id_P the axiom rule between P and $P^\perp$, of type P.

In full details, a type should contain a signature under which the formula is closed and well typed. That extra level of information would be heavy, and no real difficulty lies in dealing with it, and so we prefer to leave it implicit.

If X is a set of proofs, we shall write $\Pi : P \in X$ as a shortcut for "$\Pi \in X$ and Π has type P". We say that Π and Π' are *compatible* if their types are dual of each other.

Definition 3.6 *Orthogonality*. For $\Pi, \Pi' \in \mathcal{WN}$, we say that $\Pi \perp\!\!\!\perp \Pi'$ when for any θ and θ' such that $\Pi\theta$ and $\Pi'\theta'$ are compatible, $cut(\Pi\theta; \Pi'\theta') \in \mathcal{WN}$. For $\Pi \in \mathcal{WN}$ and $X \subseteq \mathcal{WN}$, $\Pi \perp\!\!\!\perp X$ iff $\Pi \perp\!\!\!\perp \Pi'$ for any $\Pi' \in X$, and $X^\perp$ is $\{\, \Pi \in \mathcal{WN} \ : \ \Pi \perp\!\!\!\perp X \,\}$. Finally, for $X, Y \subseteq \mathcal{WN}$, $X \perp\!\!\!\perp Y$ iff $\Pi \perp\!\!\!\perp \Pi'$ for any $\Pi \in X$, $\Pi' \in Y$.

Definition 3.7 *Reducibility candidate*. A *reducibility candidate X* is a set of normalizing proofs that is equal to its biorthogonal, that is, $X = X^{\perp\perp}$.

That kind of construction has some well-known properties,[5] which do not rely on the definition of the relation $\perp\!\!\!\perp$. For any sets of normalizable derivations X and Y, $X \subseteq Y$ implies $Y^\perp \subseteq X^\perp$ and $(X \cup Y)^\perp = X^\perp \cap Y^\perp$; moreover, the symmetry of $\perp\!\!\!\perp$ implies that $X \subseteq X^{\perp\perp}$, and hence $X^\perp = X^{\perp\perp\perp}$ (in other words, $X^\perp$ is always a candidate).

Reducibility candidates, ordered by inclusion, form a complete lattice: given an arbitrary collection of candidates S, it is easy to check that $(\bigcup S)^{\perp\perp}$ is its least upper bound in the lattice, and $\bigcap S$ its greatest lower bound. We check the minimality of $(\bigcup S)^{\perp\perp}$: any upper bound Y satisfies $\bigcup S \subseteq Y$, and hence $(\bigcup S)^{\perp\perp} \subseteq Y^{\perp\perp} = Y$. Concerning the greatest lower bound, the only nontrivial thing is that it is a candidate, but it suffices to observe that $\bigcap S = \bigcap_{X \in S} X^{\perp\perp} = (\bigcup_{X \in S} X^\perp)^\perp$. The least candidate is $\emptyset^{\perp\perp}$ and the greatest is $\mathcal{WN}$. Having a complete lattice, we can use the Knaster-Tarski theorem: any monotonic operator ϕ on reducibility candidates admits a least fixed point $\mathrm{lfp}(\phi)$ in the lattice of candidates.

Our definition of $\perp\!\!\!\perp$ yields some basic observations about candidates. They are closed under substitution, that is, $\Pi \in X$ implies that any $\Pi\theta \in X$. Indeed, $\Pi \in X$ is equivalent to $\Pi \perp\!\!\!\perp X^\perp$ which implies $\Pi\theta \perp\!\!\!\perp X^\perp$ by definition of $\perp\!\!\!\perp$ and Proposition 3.3. Hence, Id_P belongs to any candidate, since for any $\Pi \in X^\perp$, $cut(Id_{P\theta}; \Pi\theta') \to \Pi\theta' \in X^\perp \subseteq \mathcal{WN}$. Candidates are also closed under expansion, that is, $\Pi' \to \Pi$ and $\Pi \in X$ imply that $\Pi' \in X$. Indeed, for any $\Xi \in X^\perp$, $cut(\Pi'\theta; \Xi\theta') \to^* cut(\Pi\theta; \Xi\theta')$ by Proposition 3.3, and the latter derivation normalizes.

A useful simplification follows from those properties: for a candidate X, $\Pi \perp\!\!\!\perp X$ if for any θ and compatible $\Pi' \in X$, $cut(\Pi\theta; \Pi')$ normalizes; there is no need to explicitly consider instantiations of members of X, and since $Id \in X$, there is no need to show that Π normalizes by Proposition 3.4.

The generalization over all substitutions is the only novelty in our definitions. It is there to internalize the fact that proof behaviors are essentially independent of their first-order structure. By taking this into account from the beginning in the definition of orthogonality, we obtain biorthogonals (behaviors) that are closed under inessential transformations like substitution. As a result, unlike in most candidate of reducibility arguments, our candidates are untyped. In fact, we could type them up-to first-order details, that is, restrict to sets of proofs whose types have the same propositional structure. Although that might look more familiar, we prefer to avoid those unnecessary details.

Definition 3.8 *Reducibility*. Let Π be a proof of $\vdash P_1, \ldots, P_n$, and $(X_i)_{i=1\ldots n}$ a collection of reducibility candidates. We say that Π is $(X_1, \ldots, X_n)$-reducible if for any θ and any derivations $(\Pi'_i : P_i\theta^\perp \in X_i^\perp)_{i=1\ldots n}$, the derivation $cut(\Pi\theta; \Pi'_1, \ldots, \Pi'_n)$ normalizes.

---

[5]This so-called *polar* construction is used independently for reducibility candidates and phase semantics in Girard [1987], but also, for example, to define behaviors in ludics [Girard 2001].

From this definition, it immediately follows that if Π is $(X_1, \ldots, X_n)$-reducible then so is $\Pi\theta$. Also observe that Id_P is $(X, X^\perp)$-reducible for any candidate X, since for any $\Pi \in X$ and $\Pi' \in X^\perp$ $cut(Id_{P\theta}; \Pi, \Pi')$ reduces to $cut(\Pi; \Pi')$ which normalizes. Finally, any $(X_1, \ldots, X_n)$-reducible derivation Π normalizes, by Proposition 3.4 and the fact that $cut(\Pi; Id, \ldots, Id)$ normalizes.

PROPOSITION 3.9. *Let Π be a proof of $\vdash P_1, \ldots, P_n$, let $(X_i)_{i=1\ldots n}$ be a family of candidates, and let j be an index in $1 \ldots n$. The two following statements are equivalent:* (1) Π *is* $(X_1, \ldots, X_n)$*-reducible;* (2) *for any θ and* $(\Pi'_i : P_i\theta^\perp \in X_i^\perp)_{i \neq j}$, $cut(\Pi\theta; (\Pi'_i)_{i \neq j}) \in X_j$.

PROOF. (1) $\Rightarrow$ (2): Given such θ and $(\Pi'_i)_{i \neq j}$, we show that the derivation $cut(\Pi\theta; (\Pi'_i)_{i \neq j}) \in X_j$. Since $X_j = X_j^{\perp\perp}$, it is equivalent to show that our derivation is in the orthogonal of $X_j^\perp$. For each σ and $\Pi'' : P_j\theta\sigma^\perp \in X_j^\perp$, we have to show that $cut(cut(\Pi\theta; (\Pi'_i)_{i \neq j})\sigma; \Pi'')$ normalizes. Using cut permutation reductions, we reduce it into $cut(\Pi\theta\sigma; \Pi'_1\sigma, \ldots, \Pi'', \ldots, \Pi'_n\sigma)$, which normalizes by reducibility of Π. (2) $\Rightarrow$ (1) is similar: we have to show that $cut(\Pi\theta; \Pi'_1, \ldots \Pi'_n)$ normalizes, we reduce it into $cut(cut(\Pi\theta; (\Pi'_i)_{i \neq j}); \Pi'_j)$ which normalizes since $\Pi'_j \in X_j^\perp$ and the left subderivation belongs to X_j by hypothesis.   □

3.3 Interpretation

We interpret formulas as reducibility candidates, extending Girard's interpretation of MALL connectives [Girard 1987].

Definition 3.10 *Interpretation*. Let P be a formula and $\mathcal{E}$ an environment mapping each *n*-ary predicate variable p occurring in P to a candidate. We define by induction on P a candidate called *interpretation of P under $\mathcal{E}$* and denoted by $[P]^\mathcal{E}$.

$$[p\vec{t}]^\mathcal{E} = \mathcal{E}(p) \quad [a\vec{u}]^\mathcal{E} = \left\{ \overline{\vdash a\vec{v}^\perp, a\vec{v}} \right\}^{\perp\perp} \quad [\mathbf{0}]^\mathcal{E} = \emptyset^{\perp\perp} \quad [\mathbf{1}]^\mathcal{E} = \left\{ \overline{\vdash \mathbf{1}} \right\}^{\perp\perp}$$

$$[P \otimes P']^\mathcal{E} = \left\{ \frac{\dfrac{\Pi}{\vdash \Delta, Q} \quad \dfrac{\Pi'}{\vdash \Delta', Q'}}{\vdash \Delta, \Delta', Q \otimes Q'} \;:\; \Pi : Q \in [P]^\mathcal{E}, \Pi' : Q' \in [P']^\mathcal{E} \right\}^{\perp\perp}$$

$$[P_0 \oplus P_1]^\mathcal{E} = \left\{ \frac{\dfrac{\Pi}{\vdash \Delta, Q_i}}{\vdash \Delta, Q_0 \oplus Q_1} \;:\; i \in \{0, 1\}, \Pi : Q_i \in [P_i]^\mathcal{E} \right\}^{\perp\perp}$$

$$[\exists x.\, Px]^\mathcal{E} = \left\{ \frac{\dfrac{\Pi}{\vdash \Gamma, Qt}}{\vdash \Gamma, \exists x.\, Qx} \;:\; \Pi : Qt \in [Pt]^\mathcal{E} \right\}^{\perp\perp}$$

$$[u = v]^\mathcal{E} = \left\{ \overline{\vdash t = t} \right\}^{\perp\perp}$$

$$[\mu B\vec{t}]^\mathcal{E} = \mathrm{lfp}(X \mapsto \{ \mu\Pi \;:\; \Pi : B(\mu B)\vec{t} \in [Bp\vec{t}]^{\mathcal{E}, p \mapsto X} \}^{\perp\perp})$$

$$[P]^\mathcal{E} = ([P^\perp]^\mathcal{E})^\perp \text{ for all other cases}$$

The validity of that definition relies on a few observations. It is easy to check that we do only form (bi-)orthogonals of sets of proofs that are normalizing. More importantly, the existence of least fixed point candidates relies on the monotonicity of interpretations, inherited from that of operators. More generally, $[P]^\mathcal{E}$ is monotonic in $\mathcal{E}(p)$ if p occurs only positively in P, and antimonotonic in $\mathcal{E}(p)$ if p occurs only negatively. The two statements are proved simultaneously, following the definition by induction on P. Except for the least fixed point case, it is trivial to check that (anti)monotonicity

is preserved by the first clauses of Definition 3.10, and in the case of the last clause $[P]^{\mathcal{E}} = ([P^{\perp}]^{\mathcal{E}})^{\perp}$ each of our two statements is derived from the other. Let us now consider the definition of $[\mu B \vec{t}]^{\mathcal{E}}$, written $\mathrm{lfp}(\phi_{\mathcal{E}})$ for short. First, the construction is well-defined: by induction hypothesis and monotonicity of B, $[Bq\vec{t}]^{\mathcal{E}, q \mapsto X}$ is monotonic in X, and hence $\phi_{\mathcal{E}}$ is also monotonic and admits a least fixed point. We then show that $\mathrm{lfp}(\phi_{\mathcal{E}})$ is monotonic in $\mathcal{E}(p)$ when p occurs only positively in B; antimonotonicity would be obtained in a symmetric way. If $\mathcal{E}$ and $\mathcal{E}'$ differ only on p and $\mathcal{E}(p) \subseteq \mathcal{E}'(p)$, we obtain by induction hypothesis that $\phi_{\mathcal{E}}(X) \subseteq \phi_{\mathcal{E}'}(X)$ for any candidate X, and in particular $\phi_{\mathcal{E}}(\mathrm{lfp}(\phi_{\mathcal{E}'})) \subseteq \phi_{\mathcal{E}'}(\mathrm{lfp}(\phi_{\mathcal{E}'})) = \mathrm{lfp}(\phi_{\mathcal{E}'})$, that is, $\mathrm{lfp}(\phi_{\mathcal{E}'})$ is a prefixed point of $\phi_{\mathcal{E}}$, and thus $\mathrm{lfp}(\phi_{\mathcal{E}}) \subseteq \mathrm{lfp}(\phi_{\mathcal{E}'})$, that is to say $[\mu B \vec{t}]^{\mathcal{E}}$ is monotonic in $\mathcal{E}(p)$.

PROPOSITION 3.11. *For any P and $\mathcal{E}$, $([P]^{\mathcal{E}})^{\perp} = [P^{\perp}]^{\mathcal{E}}$.*

PROPOSITION 3.12. *For any P, θ and $\mathcal{E}$, $[P]^{\mathcal{E}} = [P\theta]^{\mathcal{E}}$.*

PROPOSITION 3.13. *For any $\mathcal{E}$, monotonic B and S, $[BS]^{\mathcal{E}} = [Bp]^{\mathcal{E}, p \mapsto [S]^{\mathcal{E}}}$.*

Those three propositions are easy to prove, the first one immediately following from Definition 3.10 by involutivity of both negations (on formulas and on candidates), the other two by induction (respectively on P and B). Proposition 3.12 has an important consequence: $\Pi \in [P]$ implies $\Pi\theta \in [P\theta]$, that is, our interpretation is independent of first-order aspects. This explains some probably surprising parts of the definition such as the interpretation of least fixed points, where it seems that we are not allowing the parameter of the fixed point to change from one instance to its recursive occurrences.

In the following, when the term structure is irrelevant or confusing, we shall write $[S]^{\mathcal{E}}$ for $[S\vec{t}]^{\mathcal{E}}$. For a predicate operator expression $(\lambda \vec{p}.\ B\vec{p})$ of first-order arity 0, we shall write $[B]^{\mathcal{E}}$ for $\vec{X} \mapsto [B\vec{p}]^{\mathcal{E}, (p_i \mapsto X_i)_i}$. When even more concision is desirable, we may also write $[B\vec{X}]^{\mathcal{E}}$ for $[B]^{\mathcal{E}} \vec{X}$. Finally, we simply write $[P]$ and $[B]$ when $\mathcal{E}$ is empty.

LEMMA 3.14. *Let X and Y be two reducibility candidates, and Π be a proof of $\vdash P\vec{x}, Q\vec{x}$ that is (X, Y)-reducible. Then $F_B(\Pi)$ is $([B]X, [\overline{B}]Y)$-reducible.*

LEMMA 3.15. *Let X be a candidate and Θ a derivation of $\vdash S\vec{x}^{\perp}, \overline{B}S\vec{x}$ that is $(X^{\perp}, [\overline{B}]X)$-reducible. Then $\nu(Id_{S\vec{t}}, \Theta)$ is $(X^{\perp}, [\nu\overline{B}\ \vec{t}])$-reducible for any $\vec{t}$.*

PROOF OF LEMMAS 3.14 AND 3.15. We prove them simultaneously, generalized as follows for any monotonic operator B of second-order arity $n + 1$, and any predicates $\vec{A}$ and candidates $\vec{Z}$:

(1) For any (X, Y)-reducible Π, $F_{B\vec{A}}(\Pi)$ is $([B]\vec{Z} X, [\overline{B}]\vec{Z}^{\perp} Y)$-reducible.
(2) For any $(X^{\perp}, [\overline{B}]\vec{Z}^{\perp} X)$-reducible Θ, $\nu(Id_{S\vec{t}}, \Theta)$ is $(X^{\perp}, [\nu(\overline{B}\vec{Z}^{\perp})\vec{t}])$-reducible.

We proceed by induction on B: we first establish (1), relying on strictly smaller instances of both (1) and (2); then we prove (2) by relying on (1) for the same B (modulo size-preserving first-order details). The purpose of the generalization is to separate the

main part of B from auxiliary parts $\vec{A}$, which may be large and whose interpretations $\vec{Z}$ may depend on X and Y, but play a trivial role.

(1) If B is of the form $(\lambda\vec{p}\lambda q.\ B'\vec{p})$, then $F_{B\vec{A}}(\Pi)$ is simply $Id_{B'\vec{A}}$, which is trivially $([B'\vec{Z}], [\overline{B'}\vec{Z}^\perp])$-reducible since $[\overline{B'}\vec{Z}^\perp] = [B'\vec{Z}]^\perp$. If B is of the form $(\lambda\vec{p}\lambda q.\ q\vec{t})$, then $F_{B\vec{A}}(\Pi)$ is $\Pi[\vec{t}/\vec{x}]$ which is (X, Y)-reducible.

Otherwise, B starts with a logical connective. Following the definition of F_B, dual connectives are treated in a symmetric way. The tensor case essentially consists in showing that if $\Pi' \vdash\ P', Q'$ is $([P'], [Q'])$-reducible and $\Pi'' \vdash\ P'', Q''$ is $([P''], [Q''])$-reducible then the following derivation is $([P' \otimes P''], [Q'\ \invamp\ Q''])$-reducible:

$$\frac{\dfrac{\Pi'}{\vdash P', Q'}\quad\dfrac{\Pi''}{\vdash P'', Q''}}{\dfrac{\vdash P' \otimes P'', Q', Q''}{\vdash P' \otimes P'', Q'\ \invamp\ Q''}\ \invamp}\ \otimes$$

— The subderivation $\Pi' \otimes \Pi''$ is $([P' \otimes P''], [Q'], [Q''])$-reducible: By Proposition 3.9 it suffices to show that for any θ and compatible $\Xi' \in [Q']^\perp$ and $\Xi'' \in [Q'']^\perp$, $cut(\Pi\theta; \Xi', \Xi'')$ belongs to $[P' \otimes P'']$. This follows from: the fact that it reduces to $cut(\Pi'\theta; \Xi') \otimes cut(\Pi''\theta; \Xi'')$; that those two conjuncts are respectively in $[P']$ and $[P'']$ by hypothesis; and that $\{\ u \otimes v\ :\ u \in [P'], v \in [P'']\ \}$ is a subset of $[P' \otimes P'']$ by definition of the interpretation.

— We then prove that the full derivation, instantiated by θ and cut against any compatible $\Xi \in [P' \otimes P'']^\perp$, is in $[Q'\ \invamp\ Q'']$. Since the interpretation of $\invamp$ is $\{\ u \otimes v\ :\ u \in [Q']^\perp, v \in [Q'']^\perp\ \}^\perp$, it suffices to show that $cut((\invamp(\Pi' \otimes \Pi''))\theta; \Xi)$ normalizes (which follows from the reducibility of $\Pi' \otimes \Pi''$) and that for any substitutions σ and σ', $cut((\invamp(\Pi' \otimes \Pi''))\theta; \Xi)\sigma$ normalizes when cut against any such compatible $(u \otimes v)\sigma'$. Indeed, that cut reduces, using cut permutations and the main multiplicative reduction, into $cut(cut((\Pi' \otimes \Pi'')\theta\sigma; \Xi\sigma); u\sigma', v\sigma')$ which normalizes by reducibility of $\Pi' \otimes \Pi''$.

The additive case follows the same outline. There is no case for units, including $=$ and $\neq$, since they are treated with all formulas where p does not occur.

In the case of first-order quantifiers, say $B = \lambda\vec{p}\lambda q.\ \exists x.\ B'\vec{p}qx$, we essentially have to show that, assuming that Π is $([Px], [Qx])$-reducible, the following derivation is $([\exists x.\ Px], [\forall x.\ Qx])$-reducible.

$$\frac{\dfrac{\dfrac{\Pi}{\Sigma, x; \vdash Px, Qx}}{\Sigma, x; \vdash \exists x.\ Px, Qx}\ \exists}{\Sigma; \vdash \exists x.\ Px, \forall x.\ Qx}\ \forall$$

— We first establish that the immediate subderivation $\exists(\Pi)$ is reducible, by considering $cut(\exists(\Pi)\theta; \Xi)$ for any θ and compatible $\Xi \in [Qx]^\perp$. We reduce that derivation into $\exists(cut(\Pi\theta; \Xi))$ and conclude by definition of $[\exists x.\ Px]$ and the fact that $cut(\Pi\theta; \Xi) \in [Px]$.

— To prove that $\forall(\exists(\Pi))$ is reducible, we show that $cut(\forall(\exists(\Pi))\theta; \Xi)$ belongs to $[\forall x.\ Qx]$ for any θ and compatible $\Xi \in [\exists x.\ Px]^\perp$. Since $[\forall x.\ Qx] = \{\ \exists\Xi'\ :\ \Xi' \in [Qt]^\perp\ \}^\perp$, this amounts to show that our derivation normalizes (which follows from the reducibility of $\exists(\Pi)$) and that $cut(cut(\forall(\exists(\Pi))\theta; \Xi)\sigma; (\exists\Xi')\sigma')$ normalizes for any σ, σ' and compatible $\Xi' \in [Qt]^\perp$. Indeed, this derivation reduces, by permuting the cuts and performing the main $\forall/\exists$ reduction, into $cut(\exists(\Pi)\theta\sigma[t\sigma'/x]; \Xi'\sigma', \Xi\sigma)$, which normalizes by reducibility of $\exists(\Pi)$.

Finally, we show the fixed point case in full details since this is where the generalization is really useful. When B is of the form $\lambda\vec{p}\lambda q.\ \mu(B'\vec{p}q)\vec{t}$, we are considering the following derivation.

$$
\cfrac{
\cfrac{}{\vdash \mu(B'\vec{A}P)\vec{t},\nu(\overline{B'}\vec{A}^{\perp}P^{\perp})\vec{t}}\ init
\quad
\cfrac{
\cfrac{F_{B'\vec{A}\bullet(\mu(B'\vec{A}P))\vec{x}}(\Pi)}{\vdash B'\vec{A}P(\mu(B'\vec{A}P))\vec{x},\overline{B'}\vec{A}^{\perp}Q(\nu(\overline{B'}\vec{A}^{\perp}P^{\perp}))\vec{x}}
}{\vdash \mu(B'\vec{A}P)\vec{x},\overline{B'}\vec{A}^{\perp}Q(\nu(\overline{B'}\vec{A}^{\perp}P^{\perp}))\vec{x}}\ \mu
}{\vdash \mu(B'\vec{A}P)\vec{t},\nu(\overline{B'}\vec{A}^{\perp}Q)\vec{t}}\ \nu
$$

We apply induction hypothesis (1) on $B'' := (\lambda\vec{p}\lambda p_{n+1}\lambda q.\ B'\vec{p}qp_{n+1}\vec{x})$, with $A_{n+1} := \mu(B'\vec{A}P)$ and $Z_{n+1} := [\mu(B'\vec{Z}X)]$, obtaining that the subderivation $F_{...}(\Pi)$ is $([B'']\vec{Z}ZZ_{n+1}X,[\overline{B''}]\vec{Z}^{\perp}Z_{n+1}^{\perp}Y)$-reducible. Then, we establish that $\mu(F_{...}(\Pi))$ is reducible: for any θ and compatible $\Xi \in [B'']\vec{Z}ZZ_{n+1}Y^{\perp}$, $cut(\mu(F_{...}(\Pi))\theta;\Xi)$ reduces to $\mu(cut(F_{...}(\Pi)\theta;\Xi))$ which belongs to $[\mu(B'\vec{Z}X)\vec{x}] = \{\ \mu\Pi'\ :\ \Pi' \in [B'\vec{Z}X(\mu(B'\vec{Z}X))\vec{x}]\ \}^{\perp\perp}$ by reducibility of $F_{...}(\Pi)$. We finally obtain the reducibility of the whole derivation by applying induction hypothesis (2) on B' with $A_{n+1} := Q^{\perp}$, $Z_{n+1} := Y^{\perp}$ and $X := [\mu(B'\vec{Z}X)\vec{x}]^{\perp}$.

(2) Here we have to show that for any θ and any compatible $\Xi \in X$, the derivation $cut(\nu(Id_{S\vec{t}},\Theta)\theta;\Xi)$ belongs to $[\mu(B\vec{Z})\vec{t}]^{\perp}$. Since only $\vec{t}$ is affected by θ in such derivations, we generalize on it directly, and consider the following set:

$$Y := \{\ cut(\nu(Id_{S\vec{t'}},\Theta);\Xi)\ :\ \Xi : S\vec{t'} \in X\ \}^{\perp}.$$

Note that we can form the orthogonal to obtain Y, since we are indeed considering a subset of $\mathcal{WN}$: any $cut(\nu(Id;\Theta);\Xi)$ reduces to $\nu(\Xi;\Theta)$, and Ξ and Θ normalize. We shall establish that Y is a pre-fixed point of the operator ϕ such that $[\mu(B\vec{Z})\vec{t}]$ has been defined as $\mathrm{lfp}(\phi)$, from which it follows that $[\mu(B\vec{Z})\vec{t}] \subseteq Y$, which entails our goal; note that this is essentially a proof by induction on $[\mu(B\vec{Z})]$.

So we prove the pre-fixed point property:

$$\{\ \mu\Pi\ :\ \Pi : B\vec{A}(\mu(B\vec{A}))\vec{t'} \in [B\vec{Z}Y\vec{t'}]\ \}^{\perp\perp} \subseteq Y.$$

Observing that, for any $A,B \subseteq \mathcal{WN}$, we have $A^{\perp\perp} \subseteq B^{\perp} \Leftrightarrow A^{\perp\perp} \perp\!\!\!\perp B \Leftrightarrow B \subseteq A^{\perp} \Leftrightarrow B \perp\!\!\!\perp A$, our property can be rephrased equivalently:

$$\{\ cut(\nu(Id_{S\vec{t'}},\Theta);\Xi)\ :\ \Xi : S\vec{t'} \in X\ \} \perp\!\!\!\perp \{\ \mu\Pi\ :\ \Pi \in [B\vec{Z}Y\vec{t'}]\ \}.$$

Since both sides are stable by substitution, there is no need to consider compatibility substitutions here, and it suffices to consider cuts between any compatible left and right-hand side derivations: $cut(cut(\nu(Id,\Theta);\Xi);\mu\Pi)$. It reduces, using cut exchange, the main fixed point reduction and finally the identity reduction, into the following.

$$
\cfrac{
\cfrac{
\Xi
}{\vdash \Gamma,S\vec{t'}}
\quad
\cfrac{
\cfrac{\Theta[\vec{t'}/\vec{x}]}{\vdash S^{\perp}\vec{t'},\overline{B}\vec{A}^{\perp}S\vec{t'}}
\quad
\cfrac{
\cfrac{F_{B\vec{A}\bullet\vec{t'}}(\nu(Id_{S\vec{x}},\Theta))}{\vdash B\vec{A}S^{\perp}\vec{t'},\overline{B}\vec{A}^{\perp}(\nu(\overline{B}\vec{A}^{\perp}))\vec{t'}}
\quad
\cfrac{\Pi}{\vdash B\vec{A}(\mu(B\vec{A}))\vec{t'},\Delta}
}{\vdash B\vec{A}S^{\perp}\vec{t'},\Delta}\ cut
}{\vdash S^{\perp}\vec{t'},\Delta}\ cut
}{\vdash \Gamma,\Delta}\ cut
$$

By hypothesis, $\Xi \in X$, $\Pi \in [B\vec{Z}Y\vec{t'}]$ and $\Theta[\vec{t'}/\vec{x}]$ is $(X^{\perp},[\overline{B}\vec{Z}^{\perp}X\vec{t'}])$-reducible. Moreover, $\nu(Id_{S\vec{x}},\Theta)$ is $(X^{\perp},Y^{\perp})$-reducible by definition of Y, and thus, by applying (1) on the operator $\lambda\vec{p}\lambda q.\ B\vec{p}q\vec{t'}$, which has the same size as B, we obtain that

$F_{B\vec{A}\bullet\vec{t'}}(\nu(Id_{S\vec{x}}, \Theta))$ is $([B\vec{Z}X^{\perp}\vec{t'}], [\overline{B}\vec{Z}^{\perp}Y^{\perp}\vec{t'}])$-reducible.[6] We can finally compose all that to conclude that our derivation normalizes. □

3.4 Normalization

LEMMA 3.16. *Any proof of* $\vdash P_1, \ldots, P_n$ *is* $([P_1], \ldots, [P_n])$-*reducible.*

PROOF. By induction on the height of the derivation Π, with a case analysis on the first rule. We are establishing that for any θ and compatible $(\gamma_i \in [P_i]^{\perp})_{i=1\ldots n}$, $cut(\Pi\theta; \vec{\gamma})$ normalizes. If $\Pi\theta$ is an axiom on $P \equiv P_1\theta \equiv P_2^{\perp}\theta$, the cut against a proof of $[P]$ and a proof of $[P]^{\perp}$ reduces into a cut between those two proofs, which normalizes. If $\Pi\theta = cut(\Pi'\theta; \Pi''\theta)$ is a cut on the formula P, $cut(\Pi\theta; \vec{\gamma})$ reduces to $cut(cut(\Pi'\theta; \vec{\gamma}'); cut(\Pi''\theta; \vec{\gamma}''))$ and the two subderivations belong to dual candidates $[P]$ and $[P]^{\perp}$ by induction hypothesis and Proposition 3.9.

Otherwise, Π starts with a rule from the logical group, the end sequent is of the form $\vdash \Gamma, P$ where P is the principal formula, and we shall prove that $cut(\Pi\theta; \vec{\gamma}) \in [P]$ when $\vec{\gamma}$ is taken in the duals of the interpretations of $\Gamma\theta$, which allows to conclude again using Proposition 3.9.

— The rules **1**, $\otimes$, $\oplus$, $\exists$, = and μ are treated similarly, the result coming directly from the definition of the interpretation.
Let us consider, for example, the fixed point case: $\Pi = \mu\Pi'$. By induction hypothesis, $cut(\Pi'\theta; \vec{\gamma}) \in [B(\mu B)\vec{t}]$. By definition, $[\mu B\vec{t}] = \text{lfp}(\phi) = \phi(\text{lfp}(\phi)) = X^{\perp\perp}$ where $X := \{ \mu\Xi : \Xi \in [B\bullet\vec{t}][\mu B] \}$. Since $[B(\mu B)\vec{t}] = [B\bullet\vec{t}][\mu B]$, we obtain that $\mu(cut(\Pi'\theta; \vec{\gamma})) \in X$ and thus also in $X^{\perp\perp}$. Hence $cut(\Pi\theta; \vec{\gamma})$, which reduces to the former, is also in $[\mu B\vec{t}]$.
— The rules $\perp$, $\wp$, $\top$, &, $\forall$, $\neq$, and ν are treated similarly: we establish that $cut(\Pi\theta; \vec{\gamma}) \perp\!\!\!\perp X$ for some X such that $[P] = X^{\perp}$. First, we have to show that our derivation normalizes, which comes by permuting up the cuts, and concluding by induction hypothesis — this requires that after the permutation the derivations $\vec{\gamma}$ are still in the right candidates, which relies on closure under substitution and hence signature extension for the case of disequality and $\forall$. Then we have to show that for any σ and σ', and any compatible $\Xi \in X$, the derivation $cut(cut(\Pi\theta; \vec{\gamma})\sigma; \Xi\sigma')$ normalizes too. We detail this last step for two key cases.
In the $\forall$ case we have $[\forall x. Px] = \{ \exists\Xi' : \Xi' \in [Pt^{\perp}] \}^{\perp}$, so we consider $cut(cut((\forall\Pi')\theta; \vec{\gamma})\sigma; (\exists\Xi')\sigma')$, which reduces to $cut(\Pi'\theta[t/x]; \vec{\gamma}\sigma, \Xi\sigma')$. This normalizes by induction hypothesis on $\Pi'[t/x]$, which remains smaller than Π.
The case of ν is the most complex, but is similar to the argument developed for Lemma 3.15. If Π is of the form $\nu(\Pi', \Theta)$ and $P \equiv \nu B\vec{t}$ then $cut(\Pi; \gamma)\theta$ has type $\nu B\vec{u}$ for $u := \vec{t}\theta$. Since $[\nu B\vec{u}] = \{ \mu\Xi : \Xi \in [\overline{B}\bullet\vec{u}][\mu\overline{B}] \}^{\perp}$, we show that for any σ, σ' and compatible $\Xi \in [\overline{B}(\mu\overline{B})\vec{u}]$, the derivation $cut(cut(\nu(\Pi', \Theta)\theta; \vec{\gamma})\sigma; (\mu\Xi)\sigma')$ normalizes. Let $\vec{v}$ be $\vec{u}\sigma$, the derivation reduces to:

$$cut(cut(\Pi'\theta\sigma; \vec{\gamma}\sigma); cut(\Theta[\vec{v}/\vec{x}]; cut(F_{\overline{B}\bullet\vec{v}}(\nu(Id, \Theta)); \Xi\sigma')))$$

By induction hypothesis, $cut(\Pi'\theta\sigma; \vec{\gamma}\sigma) \in [S\vec{v}]$, and Θ is $([S\vec{x}]^{\perp}, [BS\vec{x}])$-reducible. By Lemmas 3.14 and 3.15 we obtain that $F_{\overline{B}\bullet\vec{v}}(\nu(Id, \Theta))$ is $([\overline{B}S^{\perp}\vec{v}], [B(\nu B)\vec{v}])$-reducible. Finally, $\Xi \in [\overline{B}(\mu\overline{B})\vec{v}]$. We conclude by composing all these reducibilities using Proposition 3.9. □

---

[6]This use of (1) involving Y is the reason why our two lemmas need to deal with arbitrary candidates and not only interpretations of formulas.

THEOREM 3.17 CUT ELIMINATION. *Any derivation can be reduced into a cut-free derivation.*

PROOF. By Lemma 3.16, any derivation is reducible, and hence normalizes.   □

The usual immediate corollary of the cut elimination result is that μMALL is consistent, since there is obviously no cut-free derivation of the empty sequent. However, note that unlike in simpler logics, cut-free derivations do not enjoy the subformula property, because of the μ and ν rules. While it is easy to characterize the new formulas that can arise from μ, nothing really useful can be said for ν, for which no nontrivial restriction is known. Hence, μMALL only enjoys restricted forms of the subformula property, applying only to (parts of) derivations that do not involve coinductions.

4. FOCUSING

Andreoli [1992] identified some important structures in linear logic, which led to the design of his focused proof system. This complete proof system for (second-order) linear logic structures proofs in stripes of *asynchronous* and *synchronous* rules. Choices in the order of application of asynchronous rules do not matter, so that the real nondeterminism lies in the synchronous phase. However, the focused system tames this nondeterminism by forcing to hereditarily chain these choices: once the focus is set on a synchronous formula, it remains on its subformulas as its connectives are introduced, and so on, to be released only on asynchronous subformulas. We refer the reader to Andreoli [1992] for a complete description of that system, but note that Figure 2, without the fixed point rules, can be used as a fairly good reminder: it follows the exact same structure, only missing the rules for exponentials.

Focusing μMALL can be approached simply by reading the focusing of second-order linear logic through the encoding of fixed points. But this naive approach yields a poorly structured system. Let us recall the second-order encoding of $\mu B \vec{t}$:

$$\forall S.\ !(\forall \vec{x}.\ BS\vec{x} \multimap S\vec{x}) \multimap S\vec{t}.$$

This formula starts with a layer of asynchronous connectives: $\forall$, $\multimap$ and ?, the dual of !. Once the asynchronous layer has been processed, the second-order eigenvariable S represents μB and one obtains unfoldings of S into BS by focusing on the pre-fixed point hypothesis. Through that encoding, one would thus obtain a system where several unfoldings necessarily require several phase alternations. This is not satisfying: the game-based reading of focusing identifies fully synchronous (positive) formulas with data types, which should be built in one step by the player, that is, in one synchronous phase. In μMALL, least fixed points over fully synchronous operators should be seen as data types. That intuition, visible in previous examples, is also justified by the classification of connectives in Definition 2.11, and is indeed accounted for in the focused system presented in Figure 2.

It is commonly believed that asynchrony corresponds to invertibility. The two notions do coincide in many cases but it should not be taken too seriously, since this does not explain, for example, the treatment of exponentials, or the fact that *init* has to be synchronous while it is trivially invertible. In the particular case of fixed points, invertibility is of no help in designing a complete focused proof system. Both μ and ν are invertible (in the case of ν, this is obtained by using the unfolding coinvariant) but this does not capture the essential aspect of fixed points, that is their infinite behavior. As a result, a system requiring that the μ rule is applied whenever possible would not be complete, notably failing on $\vdash \top \otimes \mathbf{1}, \mu p.p$ or $\vdash nat\ x \multimap nat\ x$. As we shall see, the key to obtaining focused systems is to consider the permutability of asynchronous rules, rather than their invertibility, as the fundamental guiding principle.

Asynchronous phase

$$\dfrac{\vdash \Gamma \Uparrow P, Q, \Delta}{\vdash \Gamma \Uparrow P \,\mathbin{\mathcal{B}}\, Q, \Delta} \qquad \dfrac{\vdash \Gamma \Uparrow P, \Delta \quad \vdash \Gamma \Uparrow Q, \Delta}{\vdash \Gamma \Uparrow P \,\&\, Q, \Delta} \qquad \dfrac{\vdash \Gamma, (a^{\perp}\vec{t})^{*} \Uparrow \Delta}{\vdash \Gamma \Uparrow a^{\perp}\vec{t}, \Delta}$$

$$\dfrac{\vdash \Gamma \Uparrow \Delta}{\vdash \Gamma \Uparrow \perp, \Delta} \qquad \dfrac{}{\vdash \Gamma \Uparrow \top, \Delta} \qquad \dfrac{\{\vdash \Gamma\theta \Uparrow \Delta\theta : \theta \in csu(s \doteq t)\}}{\vdash \Gamma \Uparrow s \neq t, \Delta}$$

$$\dfrac{\vdash \Gamma \Uparrow Pc, \Delta}{\vdash \Gamma \Uparrow \forall x.Px, \Delta}$$

$$\dfrac{\vdash \Gamma \Uparrow S\vec{t}, \Delta \quad \vdash \Uparrow BS\vec{x}, S\vec{x}^{\perp}}{\vdash \Gamma \Uparrow \nu B\vec{t}, \Delta} \qquad \dfrac{\vdash \Gamma, (\nu B\vec{t})^{*} \Uparrow \Delta}{\vdash \Gamma \Uparrow \nu B\vec{t}, \Delta}$$

Synchronous phase

$$\dfrac{\vdash \Gamma \Downarrow P \quad \vdash \Gamma' \Downarrow Q}{\vdash \Gamma, \Gamma' \Downarrow P \otimes Q} \qquad \dfrac{\vdash \Gamma \Downarrow P_i}{\vdash \Gamma \Downarrow P_0 \oplus P_1} \qquad \dfrac{}{\vdash (a^{\perp}\vec{t})^{*} \Downarrow a\vec{t}}$$

$$\dfrac{}{\vdash \Downarrow \mathbf{1}} \qquad \dfrac{}{\vdash \Downarrow t = t}$$

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

$$\dfrac{\vdash \Gamma \Downarrow B(\mu B)\vec{t}}{\vdash \Gamma \Downarrow \mu B\vec{t}} \qquad \dfrac{}{\vdash (\nu \overline{B}\vec{t})^{*} \Downarrow \mu B\vec{t}}$$

Switching rules (where P is synchronous, Q asynchronous)

$$\dfrac{\vdash \Gamma, P \Uparrow \Delta}{\vdash \Gamma \Uparrow P, \Delta} \qquad \dfrac{\vdash \Gamma \Downarrow P}{\vdash \Gamma, P \Uparrow} \qquad \dfrac{\vdash \Gamma \Uparrow Q}{\vdash \Gamma \Downarrow Q}$$

Fig. 2.   The μ-focused proof-system for μMALL.

We first design the μ-focused system in Section 4.1, treating μ synchronously, which is satisfying for several reasons starting with its positive nature. We show in Section 4.2 that it is also possible to consider a focused system for μMALL where ν is treated synchronously. In Section 4.3, we apply the μ-focused system to a fragment of μLJ.

4.1 A Complete μ-Focused Calculus

In this section, we call *asynchronous* (resp. *synchronous*) the negative (resp. positive) connectives of Definition 2.11 and the formulas whose top-level connective is asynchronous (resp. synchronous). Moreover, we classify nonnegated atoms as synchronous and negated ones as asynchronous. As with Andreoli's original system, this latter choice is arbitrary and can easily be changed for a case-by-case assignment [Chaudhuri et al. 2008; Miller and Saurin 2007].

We present the system in Figure 2 as a good candidate for a focused proof system for μMALL. In addition to asynchronous and synchronous formulas as defined above, focused sequents can contain *frozen formulas* P^{*} where P is an asynchronous atom or fixed point. Frozen formulas may only be found at toplevel in sequents. 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 $\vdash \Gamma \Uparrow \Delta$. In both cases, Γ and Δ are sets of formulas of disjoint locations, and Γ is a multiset of synchronous or frozen formulas. The convention on Δ is a slight departure from Andreoli's original proof system where Δ is a list: we shall emphasize the irrelevance of the order of asynchronous rules without forcing a particular, arbitrary ordering.

Although we use an explicit freezing annotation, our treatment of atoms is really the same one as Andreoli's; the notion of freezing is introduced here as a technical device for dealing precisely with fixed points, and we also use it for atoms for a more uniform presentation.

The μ-focused system extends the usual focused system for MALL. The rules for equality are not surprising, the main novelty here is the treatment of fixed points. 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 logical structure. In accordance with Definition 2.11, μ is treated during the synchronous phase and ν during the asynchronous phase.

Roughly, what the focused system implies is that if a proof involving a ν-expression proceeds by coinduction on it, then this coinduction can be done at the beginning; otherwise that formula can be ignored in the whole derivation, except for the *init* rule. The latter case is expressed by the rule which moves the greatest fixed point to the left zone, freezing it. Focusing on a μ-expression yields two choices: unfolding or applying the initial rule for fixed points. If the considered operator is fully synchronous, the focus will never be lost. For example, if *nat* is the (fully synchronous) expression $\mu N.\lambda x.\ x = 0 \oplus \exists y.\ x = s\ y \otimes N\ y$, then focusing puts a lot of structure on a proof of $\vdash \Gamma \Downarrow nat\ t$: either t is a closed term representing a natural number and Γ is empty, or $t = s^n t'$ for some $n \geq 0$ and Γ only contains $(nat\ t')^{\perp}$.

We shall now establish the completeness of our focused proof system: If the unfocused sequent $\vdash \Gamma$ is provable then so is $\vdash\Uparrow \Gamma$, and the order of application of asynchronous rules does not affect provability. From the perspective of proofs rather than provability, we are actually going to provide transformations from unfocused to focused derivations (and back) which can reorder asynchronous rules arbitrarily. However, this result cannot hold without a simple condition avoiding pathological uses of infinite branching, as illustrated with the following counterexample. The unification problem $s\ (f\ 0) \doteq f\ (s\ 0)$, where s and 0 are constants, has infinitely many solutions $[(\lambda x.\ s^n x)/f]$. Using this, we build a derivation Π_ω with infinitely many branches, each Π_n unfolding a greatest fixed point n times.

$$\Pi_0 \overset{def}{=} \dfrac{}{\vdash \nu p.p, \top}\ \top \qquad \Pi_{n+1} \overset{def}{=} \dfrac{\dfrac{\Pi_n}{\vdash \nu p.p, \top} \quad \dfrac{}{\vdash \mu p.p, \nu p.p}\ init}{\vdash \nu p.p, \top}\ \nu$$

$$\Pi_\omega \overset{def}{=} \dfrac{\Pi_0 \quad \Pi_1 \quad \dots \quad \Pi_n \quad \dots}{f; \vdash s\ (f\ 0) \neq f\ (s\ 0), \nu p.p, \top}\ \neq$$

Although this proof happens to be already in a focused form, in the sense that focusing annotations can be added in a straightforward way, the focused transformation must also provide a way to change the order of application of asynchronous rules. In particular it must allow to permute down the introduction of the first $\nu p.p$. The only reasonable way to do so is as follows, expanding Π_0 into Π_1 and then pulling down the ν rule from each subderivation, changing Π_{n+1} into Π_n:

$$\Pi_\omega \quad \rightsquigarrow \quad \dfrac{\dfrac{\Pi_\omega}{f; \vdash s\ (f\ 0) \neq f\ (s\ 0), \nu p.p, \top} \quad \dfrac{}{\vdash \mu p.p, \nu p.p}\ init}{f; \vdash s\ (f\ 0) \neq f\ (s\ 0), \nu p.p, \top}\ \nu \quad .$$

This leads to a focusing transformation that may not terminate. The fundamental problem here is that although each additive branch only finitely explores the asynchronous formula $\nu p.p$, the overall use is infinite. A solution would be to admit infinitely deep derivations, with which such infinite balancing process may have a limit.

But our goal here is to develop finite proof representations (this is the whole point of (co)induction rules) so we take an opposite approach and require a minimum amount of finiteness in our proofs.

Definition 4.1 *Quasi-finite derivation*. A derivation is said to be quasi-finite if it is cut-free, has a finite height and only uses a finite number of different coinvariants.

This condition may seem unfortunate, but it appears to be essential when dealing with transfinite proof systems involving fixed points. More precisely, it is related to the choice regarding the introduction of asynchronous fixed points, be they greatest fixed points in μ-focusing or least fixed points in ν-focusing. Note that quasi-finiteness is trivially satisfied for any cut-free derivation that is finitely branching, and that any derivation which does not involve the $\neq$ rule can be normalized into a quasi-finite one. Moreover, quasi-finiteness is a natural condition from a practical perspective, for example in the context of automated or interactive theorem proving, where $\neq$ is restricted to finitely branching instances anyway. However, it would be desirable to refine the notion of quasi-finite derivation in a way that allows cuts and is preserved by cut elimination, so that quasi-finite proofs could be considered a proper proof frag-ment. Indeed, the essential idea behind quasi-finiteness is that only a finite number of locations are explored in a proof, and the cut-free condition is only added because cut reductions do not obviously preserve this. We conjecture that a proper, self-contained notion of quasi-finite derivation can be attained, but leave this technical development to further work.

The core of the completeness proof follows Miller and Saurin [2007]. This proof technique proceeds by transforming standard derivations into a form where focused annotations can be added to obtain a focused derivation. Conceptually, focused proofs are simply special cases of standard proofs, the annotated sequents of the focused proof system being a concise way of describing their shape. The proof transformation proceeds by iterating two lemmas which perform rule permutations: the first lemma expresses that asynchronous rules can always be applied first, while the second one expresses that synchronous rules can be applied in a hereditary fashion once the focus has been chosen. The key ingredient of Miller and Saurin [2007] is the notion of focal-ization graph, analyzing dependencies in a proof and showing that there is always at least one possible focus.

In order to ease the proof, we shall consider an intermediate proof system whose rules enjoy a one-to-one correspondence with the focused rules. This involves getting rid of the cut, nonatomic axioms, and also explicitly performing freezing.

Definition 4.2 *Freezing-annotated derivation*. The freezing-annotated variant of μMALL is obtained by removing the cut rule, enriching the sequent structure with an annotation for frozen fixed points or atoms, restricting the initial rule to be applied only on frozen asynchronous formulas, and adding explicit annotation rules.

$$\frac{}{\vdash (a^\perp \vec{t})^*, a\vec{t}} \qquad \frac{}{\vdash (\nu \overline{B}\,\vec{t})^*, \mu B\vec{t}} \qquad \frac{\vdash \Gamma, (\nu B\vec{t})^*}{\vdash \Gamma, \nu B\vec{t}} \qquad \frac{\vdash \Gamma, (a^\perp \vec{t})^*}{\vdash \Gamma, a^\perp \vec{t}}$$

Atomic instances of *init* can be translated into freezing-annotated derivations.

$$\frac{}{\vdash \nu B\vec{t}, \mu \overline{B}\,\vec{t}} \quad \longrightarrow \quad \frac{\dfrac{}{\vdash (\nu B\vec{t})^*, \mu \overline{B}\,\vec{t}}}{\vdash \nu B\vec{t}, \mu \overline{B}\,\vec{t}} \qquad\qquad \frac{}{\vdash a^\perp \vec{t}, a\vec{t}} \quad \longrightarrow \quad \frac{\dfrac{}{\vdash (a^\perp \vec{t})^*, a\vec{t}}}{\vdash a^\perp \vec{t}, a\vec{t}}$$

Arbitrary instances of *init* can also be obtained by first expanding them to rely only on atomic *init*, using Proposition 2.8, and then translating atomic *init* as shown above. We shall denote by *init*∗ this derived generalized axiom. Any μMALL derivation can

be transformed into a freezing-annotated one by normalizing it and translating *init* into *init∗*.

The asynchronous freezing-annotated rules (that is, those whose principal formula is asynchronous) correspond naturally to asynchronous rules of the μ-focused system. Similarly, synchronous freezing-annotated rules correspond to synchronous focused rules, which includes the axiom rule. The switching rules of the μ-focused system do not have a freezing-annotated equivalent: they are just book-keeping devices marking phase transitions.

From now on we shall work on freezing-annotated derivations, simply calling them derivations.

4.1.1 Balanced Derivations. In order to ensure that the focalization process terminates, we have to guarantee that the permutation steps preserve some measure over derivations. The main problem here comes from the treatment of fixed points, and more precisely from the fact that there is a choice in the asynchronous phase regarding greatest fixed points. We must ensure that a given greatest fixed point formula is always used in the same way in all additive branches of a proof: if a greatest fixed point is copied by an additive conjunction or $\neq$, then it should either be used for coinduction in all branches, or frozen and used for axiom in all branches. Otherwise it would not be possible to permute the treatment of the ν under that of the & or $\neq$ while controlling the size of the transformed derivation.

Definition 4.3 *Balanced derivation*. A greatest fixed point occurrence is *used in a balanced way* if all of its principal occurrences are used consistently: either they are all frozen or they are all used for coinduction, with the same coinvariant. We say that a derivation is *balanced* if it is quasi-finite and all greatest fixed points occurring in it are used in a balanced way.

LEMMA 4.4. *If S_0 and S_1 are both coinvariants for B then so is $S_0 \oplus S_1$.*

PROOF. Let Π_i be the derivation of coinvariance for S_i. The proof of coinvariance of $S_0 \oplus S_1$ is as follows.

$$\cfrac{\cfrac{\phi_0(\Pi_0)}{\vdash S_0^\perp \vec{x},\, B(S_0 \oplus S_1)\vec{x}} \quad \cfrac{\phi_1(\Pi_1)}{\vdash S_1^\perp \vec{x},\, B(S_0 \oplus S_1)\vec{x}}}{\vdash S_0^\perp \vec{x}\ \&\ S_1^\perp \vec{x},\, B(S_0 \oplus S_1)\vec{x}}\ \&$$

The transformed derivations $\phi_i(\Pi_i)$ are obtained by functoriality as follows.

$$\phi_i(\Pi_i) = \cfrac{\cfrac{\Pi_i}{\vdash S_i^\perp \vec{x},\, BS_i\vec{x}} \quad \cfrac{\cfrac{\cfrac{\overline{\vdash S_i^\perp \vec{y},\, S_i\vec{y}}\ init}{\vdash S_i^\perp \vec{y},\, S_0\vec{y} \oplus S_1\vec{y}}\ \oplus}{\vdash \overline{B}S_i^\perp \vec{x},\, B(S_0 \oplus S_1)\vec{x}}\ B}{}}{\vdash S_i^\perp \vec{x},\, B(S_0 \oplus S_1)\vec{x}}\ cut$$

Notice that after the elimination of cuts, the proof of coinvariance that we built can be larger than the original ones: this is why this transformation cannot be done as part of the rule permutation process. □

LEMMA 4.5. *Any quasi-finite derivation of $\vdash \Gamma$ can be transformed into a balanced derivation of $\vdash \Gamma$.*

PROOF. We first ensure that all coinvariants used for the same (locatively identical) greatest fixed point are the same. For each νB on which at least one coinduction is

performed in the proof, this is achieved by taking the union of all coinvariants used in the derivation, thanks to Lemma 4.4, adding to this union the unfolding coinvariant $B(\nu B)$. Note that quasi-finiteness is needed here to ensure that we are only combining finitely many coinvariants. Let $S_{\nu B}$ be the resulting coinvariant, of the form $S_0 \oplus \ldots \oplus S_n \oplus B(\nu B)$, and $\Theta_{\nu B}$ be the proof of its coinvariance. We adapt our derivation by changing every instance of the ν rule as follows.

$$
\frac{\vdash \Gamma, S_i \vec{t} \quad \dfrac{\Theta_i}{\vdash S_i^{\perp} \vec{x}, BS_i \vec{x}}}{\vdash \Gamma, \nu B \vec{t}} \qquad \longrightarrow \qquad \frac{\dfrac{\vdash \Gamma, S_i \vec{t}}{\vdash \Gamma, S_{\nu B} \vec{t}} \oplus \dfrac{\Theta_{\nu B}}{\vdash S_{\nu B}^{\perp} \vec{x}, BS_{\nu B} \vec{x}}}{\vdash \Gamma, \nu B \vec{t}}
$$

It remains to ensure that a given fixed point is either always coinducted on or always frozen in the derivation. We shall balance greatest fixed points, starting with unbalanced fixed points closest to the root, and potentially unbalancing deeper fixed points in that process, but without ever introducing unbalanced fixed points that were not initially occurring in the proof.

Let Π_0 be the derivation obtained at this point. We define the degree of a greatest fixed point to be the maximum distance in the sublocation ordering to a greatest fixed point sublocation occurring in Π_0, 0 if there is none. Quasi-finiteness ensures that degrees are finite, since there are only finitely many locations occurring at toplevel in the sequents of a quasi-finite derivation. We shall only consider derivations in which greatest fixed points that are coinducted on are also coinducted on with the same coinvariant in Π_0, and maintain this condition while transforming any such derivation into a balanced one. We proceed by induction on the multiset of the degrees of unbalanced fixed points in the derivation, ordered using the standard multiset ordering; note that degrees are well defined for all unbalanced fixed points since they must also occur in Π_0. If there is no unbalanced fixed point, we have a balanced proof. Otherwise, pick an unbalanced fixed point of maximal degree. It is frozen in some branches and coinducted on in others. We remove all applications of freezing on that fixed point, which requires to adapt axioms[7].

$$
\frac{}{\vdash (\nu B \vec{t})^{*}, \mu \overline{B}\, \vec{t}} \quad \longrightarrow \quad \frac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash B(\nu B)\vec{t}, \overline{B(\mu \overline{B})}\vec{t}}\ init*}{\vdash B(\nu B)\vec{t}, \mu \overline{B}\, \vec{t}}\ \mu}{\vdash S_{\nu B}\vec{t}, \mu \overline{B}\, \vec{t}} \oplus \dfrac{\Theta_{\nu B}}{\vdash S_{\nu B}^{\perp} \vec{x}, BS_{\nu B}\vec{x}}}{\vdash \nu B \vec{t}, \mu \overline{B}\, \vec{t}}\ \nu
$$

The fixed point νB is used in a balanced way in the resulting derivation. Our use of the derived rule $init*$ might have introduced some new freezing rules on greatest fixed point sublocations of $B(\nu B)$ or $\overline{B}(\mu \overline{B})$. Such sublocations, if already present in the proof, may become unbalanced, but have a smaller degree. Some new sublocations may also be introduced, but they are only frozen as required. The new derivation has a smaller multiset of unbalanced fixed points, and we can conclude by induction hypothesis. □

Balancing is the most novel part of our focalization process. This preprocessing is a technical device ensuring termination in the proof of completeness, whatever rule permutations are performed. It should be noted that balancing is often too strong, and that many focused proofs are indeed not balanced. For example, it is possible to obtain

---

[7]Note that instead of the unfolding coinvariant $B(\nu B)$ we could have used the coinvariant νB. This would yield a simpler proof, but that would not be so easy to adapt for ν-focusing in Section 4.2.

unbalanced focused proofs by introducing an additive conjunction before treating a greatest fixed point differently in each branch.

4.1.2 Focalization Graph. We shall now present the notion of focalization graph and its main properties [Miller and Saurin 2007]. As we shall see, their adaptation to μMALL is trivial.[8]

Definition 4.6. The *synchronous trunk* of a derivation is its largest prefix containing only applications of synchronous rules. It is a potentially open subderivation having the same conclusion sequent. The open sequents of the synchronous trunk (which are conclusions of asynchronous rules in the full derivation) and its initial sequents (which are conclusions of *init*, **1** or =) are called *leaf sequents* of the trunk.

Definition 4.7. We define the relation $\prec$ on the formulas of the base sequent of a derivation Π: $P \prec Q$ if and only if there exists P', asynchronous subformula[9] of P, and Q', synchronous subformula of Q, such that P' and Q' occur in the same leaf sequent of the synchronous trunk of Π.

The intended meaning of $P \prec Q$ 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 4.8. *If Π starts with a synchronous rule, and P is minimal for $\prec$ in Π, then so are its subformulas in their respective subderivations.*

PROOF. There is nothing to do if Π simply consists of an initial rule. In all other cases ($\otimes$, $\oplus$, $\exists$ and μ) let us consider any subderivation Π' in which the minimal element P or one of its subformulas P' occurs — there will be exactly one such Π', except in the case of a tensor applied on P. The other formulas occurring in the conclusion of Π' either occur in the conclusion of Π or are subformulas of the principal formula occurring in it. This implies that a $Q \prec P$ or $Q \prec P'$ in Π' would yield a $Q' \prec P$ in Π, which contradicts the minimality hypothesis. □

LEMMA 4.9. *The relation $\prec$ is acyclic.*

PROOF. We proceed by induction on the derivation Π. If it starts with an asynchronous rule or an initial synchronous rule, *i.e.,* its conclusion sequent is a leaf of its synchronous trunk, acyclicity is obvious since $P \prec Q$ iff P is asynchronous and Q is synchronous. If Π starts with $\oplus$, $\exists$ or μ, the relations $\prec$ in Π and its subderivation are isomorphic (only the principal formula changes) and we conclude by induction hypothesis. In the case of $\otimes$, say Π derives $\vdash \Gamma, \Gamma', P \otimes P'$, only the principal formula $P \otimes P'$ has subformulas in both premises $\vdash \Gamma, P$ and $\vdash \Gamma', P'$. Hence there cannot be any $\prec$ relation between a formula of Γ and one of Γ'. In fact, the graph of $\prec$ in the conclusion is obtained by taking the union of the graphs in the premises and merging P and P' into $P \otimes P'$. Suppose, *ab absurdo*, that $\prec$ has cycles in Π, and consider a cycle of minimal length. It cannot involve nodes from both Γ and Γ': since only $P \otimes P'$ connects those two components, the cycle would have to go twice through it, which contradicts the minimality of the cycle's length. Hence the cycle must lie within $(\Gamma, P \otimes P')$ or $(\Gamma', P \otimes P')$ but then there would also be a cycle in the corresponding

premise (obtained by replacing $P \otimes P'$ by its subformula) which is absurd by induction hypothesis. $\qquad\square$

4.1.3 Permutation Lemmas and Completeness. We are now ready to describe the transformation of a balanced derivation into a μ-focused derivation.

Definition 4.10. We define the *reachable locations* of a balanced derivation Π, denoted by $|\Pi|$, by taking the finitely many locations occurring at toplevel in sequents of Π, ignoring coinvariance subderivations, and saturating this set by adding the sublocations of locations that do not correspond to fixed point expressions.

It is easy to see that $|\Pi|$ is a finite set. Hence $|\Pi|$, ordered by strict inclusion, is a well-founded measure on balanced derivations.

Let us illustrate the role of reachable locations with the following derivations.

$$
\cfrac{\cfrac{}{\vdash S\vec{t}, a \,\bindnasrepma\, b\,, \top}\ \top \quad \cfrac{\raisebox{0.3ex}{$\vdots$}}{\vdash S^{\perp}\vec{x}, BS\vec{x}}}{\vdash \nu B\vec{t}, a \,\bindnasrepma\, b\,, \top}\ \nu
\qquad\qquad
\cfrac{\cfrac{}{\vdash \nu B\vec{t}, a, b\,, \top}\ \top}{\vdash \nu B\vec{t}, a \,\bindnasrepma\, b\,, \top}\ \bindnasrepma
$$

For the first derivation, the set of reachable locations is $\{\nu B\vec{t}, a \,\bindnasrepma\, b\,, \top, S\vec{t}, a, b\}$. For the second one, it is $\{\nu B\vec{t}, a \,\bindnasrepma\, b\,, \top, a, b\}$. As we shall see, the focalization process may involve transforming the first derivation into the second one, thus loosing reachable locations, but it will never introduce new ones. In that process, the asynchronous rule $\bindnasrepma$ is "permuted" under the $\top$, that is, the application of $\top$ is delayed by the insertion of a new $\bindnasrepma$ rule. This limited kind of proof expansion does not affect reachable locations. A more subtle case is that of "permuting" a fixed point rule under $\top$. This will never happen for μ. For ν, the permutation will be guided by the existing reachable locations: if ν currently has no reachable sublocation it will be frozen, otherwise it will be coinducted on, leaving reachable sublocations unchanged in both cases. The set of reachable locations is therefore a skeleton that guides the focusing process, and a measure which ensures its termination.

LEMMA 4.11. *For any balanced derivation Π, $|\Pi\theta|$ is balanced and $|\Pi\theta| \subseteq |\Pi|$.*

PROOF. By induction on Π, following the definition of $\Pi\theta$. The preservation of balancing and reachable locations is obvious since the rule applications in $\Pi\theta$ are the same as in Π, except for branches that are erased by θ (which can lead to a strict inclusion of reachable locations). $\qquad\square$

LEMMA 4.12 ASYNCHRONOUS PERMUTABILITY. *Let P be an asynchronous formula. If $\vdash \Gamma, P$ has a balanced derivation Π, then it also has a balanced derivation Π' where P is principal in the conclusion sequent, and such that $|\Pi'| \subseteq |\Pi|$.*

PROOF. Let Π_0 be the initial derivation. We proceed by induction on its subderivations, transforming them while respecting the balanced use of fixed points in Π_0. If P is already principal in the conclusion, there is nothing to do. Otherwise, by induction hypothesis we make P principal in the immediate subderivations where it occurs, and we shall then permute the first two rules.

If the first rule $\mathcal{R}$ is $\top$ or a nonunifiable instance of $\neq$, there is no subderivation, and *a fortiori* no subderivation where P occurs. In that case we apply an introduction rule for P, followed by $\mathcal{R}$ in each subderivation. This is obvious in the case of $\bindnasrepma$, &, $\forall$, $\bot$, $\neq$ and $\top$ (note that there may not be any subderivation in the last two cases, in which case the introduction of P replaces $\mathcal{R}$). If P is a greatest fixed point that is coinducted on in Π_0, we apply the coinduction rule with the coinvariance premise

taken in Π_0, followed by $\mathcal{R}$. Otherwise, we freeze P and apply $\mathcal{R}$. By construction, the resulting derivation is balanced in the same way as Π_0, and its reachable locations are contained in $|\Pi_0|$.

In all other cases we permute the introduction of P under the first rule. The permutations of MALL rules are simple. We shall not detail them, but note that if P is $\top$ or a nonunifiable $u \neq v$, permuting its introduction under the first rule erases that rule. The permutations involving freezing rules are obvious, and most of the ones involving fixed points, such as $\otimes/\nu$, are not surprising.

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

The $\&/\nu$ and $\neq/\nu$ permutations rely on the fact that the subderivations obtained by induction hypothesis are balanced in the same way, with one case for freezing in all additive branches and one case for coinduction in all branches.

$$\frac{\dfrac{\dfrac{\Pi}{\vdash \Gamma, P, S\vec{t}} \quad \dfrac{\Theta}{\vdash BS\vec{x}, S\vec{x}^\perp}}{\vdash \Gamma, P, \nu B\vec{t}} \quad \dfrac{\dfrac{\Pi'}{\vdash \Gamma, P', S\vec{t}} \quad \dfrac{\Theta}{\vdash BS\vec{x}, S\vec{x}^\perp}}{\vdash \Gamma, P', \nu B\vec{t}}}{\vdash \Gamma, P \,\&\, P', \nu B\vec{t}}$$

$$\downarrow$$

$$\frac{\dfrac{\dfrac{\Pi}{\vdash \Gamma, P, S\vec{t}} \quad \dfrac{\Pi'}{\vdash \Gamma, P', S\vec{t}}}{\vdash \Gamma, P \,\&\, P', S\vec{t}} \quad \dfrac{\Theta}{\vdash BS\vec{x}, (S\vec{x})^\perp}}{\vdash \Gamma, P \,\&\, P', \nu B\vec{t}}$$

Another nontrivial case is $\otimes/\neq$ which makes use of Lemma 4.11.

$$\frac{\dfrac{\left\{\dfrac{\Pi_\sigma}{\vdash (\Gamma, P)\sigma} \;:\; \sigma \in csu(u \doteq v)\right\}}{\vdash \Gamma, P, u \neq v} \quad \dfrac{\Pi'}{\vdash \Gamma', Q}}{\vdash \Gamma, \Gamma', P \otimes Q, u \neq v}$$

$$\downarrow$$

$$\frac{\left\{\dfrac{\dfrac{\Pi_\sigma}{\vdash (\Gamma, P)\sigma} \quad \dfrac{\Pi'\sigma}{\vdash (\Gamma', Q)\sigma}}{\vdash (\Gamma, \Gamma', P \otimes Q)\sigma} \;:\; \sigma \in csu(u \doteq v)\right\}}{\vdash \Gamma, \Gamma', P \otimes Q, u \neq v}$$

A simple inspection shows that in each case, the resulting derivation is balanced in the same way as Π_0, and does not have any new reachable location; the set of reachable locations may strictly decrease only upon proof instantiation in $\otimes/\neq$, or when permuting $\top$ and trivial instances of $\neq$ under other rules. $\square$

LEMMA 4.13 SYNCHRONOUS PERMUTABILITY. *Let Γ be a sequent of synchronous and frozen formulas. If $\vdash \Gamma$ has a balanced derivation Π in which P is minimal for $\prec$ then it also has a balanced derivation Π' such that P is minimal and principal in the conclusion sequent of Π', and $|\Pi'| = |\Pi|$.*

PROOF. We proceed by induction on the derivation. If P is already principal, there is nothing to do. Otherwise, since the first rule must be synchronous, P occurs in a single subderivation. We can apply our induction hypothesis on that subderivation: its conclusion sequent still cannot contain any asynchronous formula by minimality of P and, by Proposition 4.8, P is still minimal in it. We shall now permute the first two rules, which are both synchronous. The permutations of synchronous MALL rules are simple. As for $\mathbf{1}$, there is no permutation involving $=$. The permutations for μ follow the same geometry as those for $\exists$ or $\oplus$. For instance, $\otimes/\mu$ is as follows.

$$
\cfrac{\vdash \Gamma, P \quad \cfrac{\vdash \Gamma', P', B(\mu B)\vec{t}}{\vdash \Gamma', P', \mu B\vec{t}}\ \mu}{\vdash \Gamma, \Gamma', P \otimes P', \mu B\vec{t}}\ \otimes
\qquad \longrightarrow \qquad
\cfrac{\cfrac{\vdash \Gamma, P \quad \vdash \Gamma', P', B(\mu B)\vec{t}}{\vdash \Gamma, \Gamma', P \otimes P', B(\mu B)\vec{t}}\ \otimes}{\vdash \Gamma, \Gamma', P \otimes P', \mu B\vec{t}}\ \mu
$$

All those permutations preserve $|\Pi|$. Balancing and minimality are obviously preserved, respectively because asynchronous rule applications and the leaf sequents of the synchronous trunk are left unchanged. $\qquad\square$

THEOREM 4.14. *The μ-focused system is sound and complete with respect to μMALL: If $\vdash\Uparrow \Gamma$ is provable, then $\vdash \Gamma$ is provable in μMALL. If $\vdash \Gamma$ has a quasi-finite μMALL derivation, then $\vdash\Uparrow \Gamma$ has a (focused) derivation.*

PROOF. For soundness, we observe that an unfocused derivation can be obtained simply from a focused one by erasing focusing annotations and removing switching rules ($\vdash \Delta \Uparrow \Gamma$ gives $\vdash \Delta, \Gamma$ and $\vdash \Gamma \Downarrow P$ gives $\vdash \Gamma, P$). To prove completeness, we first obtain a balanced derivation using Lemma 4.5. Then, we use permutation lemmas to reorder rules in the freezing-annotated derivation so that we can translate it to a μ-focused derivation. Formally, we first use an induction on the height of the derivation. This allows us to assume that coinvariance proofs can be focused, which will be preserved since those subderivations are left untouched by the following transformations. Then, we prove simultaneously the following two statements.

(1) If $\vdash \Gamma, \Delta$ has a balanced derivation Π, where Γ contains only synchronous and frozen formulas, then $\vdash \Gamma \Uparrow \Delta$ has a derivation.
(2) If $\vdash \Gamma, P$ has a balanced derivation Π in which P is minimal for $\prec$, and there is no asynchronous formula in its conclusion, then there is a focused derivation of $\vdash \Gamma \Downarrow P$.

We proceed by well-founded induction on $|\Pi|$ with a subinduction on the number of nonfrozen formulas in the conclusion of Π. Note that (1) can rely on (2) for the same $|\Pi|$ but (2) only relies on strictly smaller instances of (1) and (2).

(1) If there is any, pick arbitrarily an asynchronous formula P, and apply Lemma 4.12 to make it principal in the first rule. The subderivations of the obtained proof can be focused, either by the outer induction in the case of coinvariance proofs, or by induction hypothesis (1) for the other subderivations: if the first rule is a freezing, then the reachable locations of the subderivation and the full derivation are the same, but there is one less nonfrozen formula; with all other rules, the principal location is consumed and reachable locations strictly decrease. Finally, we obtain the full focused derivation by composing those subderivations using the focused equivalent of the rule applied on P.
When there is no asynchronous formula left, we have shown in Lemma 4.9 that there is a minimal synchronous formula P in Γ, Δ. Let Γ' denote Γ, Δ without P.

Using switching rules, we build the derivation of $\vdash \Gamma \Uparrow \Delta$ from $\vdash \Gamma' \Downarrow P$, the latter derivation being obtained by (2) with Π unchanged.

(2) Given such a derivation, we apply Lemma 4.13 to make the formula P principal. Each of its subderivations has strictly less reachable locations, and a conclusion of the form $\vdash \Gamma''$, P' where P' is a subformula of P that is still minimal by Proposition 4.8. For each of those we build a focused derivation of $\vdash \Gamma'' \Downarrow P'$: if the subderivation still has no asynchronous formula in its conclusion, we can apply induction hypothesis (2); otherwise P' is asynchronous by minimality and we use the switching rule releasing focus on P', followed by a derivation of $\vdash \Gamma'' \Uparrow P'$ obtained by induction hypothesis (1). Finally, we build the expected focused derivation from those subderivations by using the focused equivalent of the synchronous freezing-annotated rule applied on P. □

In addition to a proof of completeness, we have actually defined a transformation that turns any unfocused proof into a focused one. This process is in three parts: first, balancing a quasi-finite unfocused derivation; then, applying rule permutations on unfocused balanced derivations; finally, adding focusing annotations to obtain a focused proof. The core permutation process allows to reorder asynchronous rules arbitrarily, establishing that, from the proof search viewpoint, this phase consists of inessential non-determinism as usual, except for the choice concerning greatest fixed points.

In the absence of fixed points, balancing disappears, and the core permutation process is known to preserve the essence of proofs, that is, the resulting derivation behaves the same as the original one with respect to cut elimination. A natural question is whether our process enjoys the same property. This is not a trivial question, because of the merging of coinvariants which is performed during balancing, and to a smaller extent the unfoldings also performed in that process. We conjecture that those new transformations, which are essentially loop fusions and unrolling, do also preserve the cut elimination behavior of proofs.

A different proof technique for establishing completeness consists in focusing a proof by cutting it against focused identities [Laurent 2004; Chaudhuri et al. 2008]. The preservation of the essence of proofs is thus an immediate corollary of that method. However, the merging of coinvariants cannot be performed through cut elimination, so this proof technique (alone) cannot be used in our case.

4.2 The ν-Focused System

While the classification of μ as synchronous and ν as asynchronous is rather satisfying and coincides with several other observations, that choice does not seem to be forced from the focusing point of view alone. After all, the μ rule also commutes with all other rules. It turns out that one can design a ν-focused system treating μ as asynchronous and ν as synchronous, and still obtain completeness. That system is obtained from the previous one by changing only the rules working on fixed points as follows.

$$\frac{\vdash \Gamma \Uparrow B(\mu B)\vec{t}, \Delta}{\vdash \Gamma \Uparrow \mu B\vec{t}, \Delta} \qquad \frac{\vdash \Gamma, (\mu B\vec{t})^* \Uparrow \Delta}{\vdash \Gamma \Uparrow \mu B\vec{t}, \Delta}$$

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

Note that a new asynchronous phase must start in the coinvariance premise: asynchronous connectives in $BS\vec{x}$ or $(S\vec{x})^\perp$ might have to be introduced before a focus can be picked. For example, if B is $(\lambda p.\ a^\perp \,\mathbin{⅋}\, \bot)$ and S is $a^\perp$, one cannot focus on $S^\perp$ immediately since $a^\perp$ is not yet available for applying the *init*; conversely, if B is $(\lambda p.\ a)$ and S is $a \otimes \mathbf{1}$, one cannot focus on BS immediately.

THEOREM 4.15. *The ν-focused system is sound and complete with respect to μMALL: If $\vdash\Uparrow \Gamma$ is provable, then $\vdash \Gamma$ is provable in μMALL. If $\vdash \Gamma$ has a quasi-finite μMALL derivation, then $\vdash\Uparrow \Gamma$ has a (focused) derivation.*

PROOF SKETCH. The proof follows the same argument as for the μ-focused system. We place ourselves in a logic with explicit freezing annotations for atoms and least fixed points, and define balanced annotated derivations, requiring that any instance of a least fixed point is used consistently throughout a derivation, either always frozen or always unfolded; together with the constraint on its sublocations, this means that a least fixed point has to be unfolded the same number of times in all (additive) branches of a derivation. We then show that any quasi-finite annotated derivation can be balanced; the proof of Lemma 4.5 can be adapted easily. Finally, balanced derivations can be transformed into focused derivations using permutations: the focalization graph technique extends trivially, the new asynchronous permutations involving the μ rule are simple thanks to balancing, and the new synchronous permutations involving the ν rule are trivial.   □

This flexibility in the design of a focusing system is unusual. It is not of the same nature as the arbitrary bias assignment that can be used in Andreoli's system: atoms are noncanonical, and the bias can be seen as a way to indicate what is the synchrony of the formula that a given atom might be instantiated with. But our fixed points have a fully defined logical meaning, they are canonical. The flexibility highlights the fact that focusing is a somewhat shallow property, accounting for local rule permutability independently of deeper properties such as positivity.

Although we do not see any practical use of such flexibility, it is not excluded that one is discovered in the future, like with the arbitrary bias assignment on atoms in Andreoli's original system.

It is not possible to treat both least and greatest fixed points as asynchronous. Besides creating an unclear situation regarding *init*, this would require to balance both kinds of fixed points, which is impossible. In μ-focusing, balancing greatest fixed points unfolds least fixed points as a side effect, which is harmless since there is no balancing constraint on those. The situation is symmetric in ν-focusing. But if both least and greatest fixed points have to be balanced, the two unfolding processes interfere and may not terminate anymore. It is nevertheless possible to consider mixed bias assignments for fixed point formulas, if the *init* rule is restricted accordingly. We would consider two logically identical variants of each fixed point: μ^+ and ν^+ being treated synchronously, μ^- and ν^- asynchronously, and the axiom rule would be restricted to dual fixed points of opposite bias.

$$\overline{\vdash (\mu B\vec{t})^+, (\nu\overline{B}\vec{t})^-} \quad \overline{\vdash (\nu B\vec{t})^+, (\mu\overline{B}\vec{t})^-}$$

This restriction allows to perform simultaneously the balancing of ν^- and μ^- without interferences. Further, we conjecture that a sound and complete focused proof system for that logic would be obtained by superposing the μ-focused system for μ^+, ν^- and the ν-focused system for μ^-, ν^+.

4.3 Application to μLJL

The examples of Section 2.6 showed 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 μLJ [Baelde 2008a], the extension of LJ with least and greatest fixed points. In the following, we show a simple first step to this program, in which we capture a rich fragment of μLJ even though μMALL does not have exponentials. In this section, we make use of the properties of negative formulas (Definition 2.11),

which has two important consequences: we shall use the μ-focused system, and could not use the alternative ν-focused one, since it does not agree with the classification; moreover, we shall work in a fragment of μMALL without atoms, since atoms do not have any polarity.

We have observed (Proposition 2.12) that structural rules are admissible for negative formulas of μMALL. This property allows us to obtain a faithful encoding of a fragment of μLJ in μMALL despite the absence of exponentials. The encoding must be organized so that formulas appearing on the left-hand side of intuitionistic sequents can be encoded positively in μMALL. The only connectives allowed to appear negatively shall 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 4.16 $\mathcal{H}$, $\mathcal{G}$, μ*LJL*. The fragments $\mathcal{H}$ and $\mathcal{G}$ are given by the following grammar.

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

The logic μLJL is the restriction of μLJ to sequents where all hypotheses are in the fragment $\mathcal{H}$, and the goal is in the fragment $\mathcal{G}$. This implies a restriction of induction and coinduction rules to (co)invariants in $\mathcal{H}$.

Formulas in $\mathcal{H}$ and $\mathcal{G}$ are translated in μMALL as follows.

$$[P \wedge Q] \stackrel{def}{=} [P] \otimes [Q] \qquad\qquad [\forall x.Px] \stackrel{def}{=} \forall x.[Px]$$
$$[P \vee Q] \stackrel{def}{=} [P] \oplus [Q] \qquad\qquad [\nu B\vec{t}] \stackrel{def}{=} \nu[B]\vec{t}$$
$$[s = t] \stackrel{def}{=} s = t \qquad\qquad [P \supset Q] \stackrel{def}{=} [P] \multimap [Q]$$
$$[\exists x.Px] \stackrel{def}{=} \exists x.[Px] \qquad\qquad [\lambda p\lambda\vec{x}.Bp\vec{x}] \stackrel{def}{=} \lambda p\lambda\vec{x}.[Bp\vec{x}]$$
$$[\mu B\vec{t}] \stackrel{def}{=} \mu[B]\vec{t} \qquad\qquad [p\vec{t}] \stackrel{def}{=} p\vec{t}$$

For reference, the rules of μLJL can be obtained simply from the rules of the focused system presented in Figure 3, by translating $\Gamma; \Gamma' \vdash P$ into $\Gamma, \Gamma' \vdash P$, allowing both contexts to contain any $\mathcal{H}$ formula and reading them as sets to allow contraction and weakening.

PROPOSITION 4.17. *Let P be a $\mathcal{G}$ formula, and Γ a context of $\mathcal{H}$ formulas. Then $\Gamma \vdash P$ has a quasi-finite μLJL derivation if and only if $\vdash [\Gamma]^{\perp}, [P]$ has a quasi-finite μMALL derivation, under the restrictions that (co)invariants in μMALL are of the form $\lambda\vec{x}. [S\vec{x}]$ for $S\vec{x} \in [\mathcal{H}]$.*

PROOF. The proof transformations are simple and compositional. The induction rule corresponds to the ν rule for $(\mu[B]\vec{t})^{\perp}$, the proviso on invariants allowing the translations.

$$\frac{\Gamma, S\vec{t} \vdash G \quad BS\vec{x} \vdash S\vec{x}}{\Gamma, \mu B\vec{t} \vdash G} \qquad \longleftrightarrow \qquad \frac{\vdash [\Gamma]^{\perp}, [S]^{\perp}\vec{t}, [G] \quad \vdash \overline{[B]}[S]^{\perp}\vec{x}, [S]\vec{x}}{\vdash [\Gamma]^{\perp}, \nu\overline{[B]}\vec{t}, [G]}$$

$$\text{Asynchronous phase}$$

$$\frac{\Gamma;\Gamma',P,Q \vdash R}{\Gamma;\Gamma',P \wedge Q \vdash R} \qquad \frac{\Gamma;\Gamma',P \vdash R \quad \Gamma;\Gamma',Q \vdash R}{\Gamma;\Gamma',P \vee Q \vdash R}$$

$$\frac{\Gamma;\Gamma',Px \vdash Q}{\Gamma;\Gamma',\exists x.Px \vdash Q}$$

$$\frac{\{(\Gamma;\Gamma' \vdash P)\theta : \theta \in csu(s \doteq t)\}}{\Gamma;\Gamma',s = t \vdash P}$$

$$\frac{\Gamma,\mu B\vec{t};\Gamma' \vdash P}{\Gamma;\Gamma',\mu B\vec{t} \vdash P} \qquad \frac{S \in \mathcal{H} \quad \Gamma;\Gamma',S\vec{t} \vdash P \quad ;BS\vec{x} \vdash S\vec{x}}{\Gamma;\Gamma',\mu B\vec{t} \vdash P}$$

$$\text{Synchronous phase}$$

$$\frac{\Gamma;\vdash A \quad \Gamma;\vdash B}{\Gamma;\vdash A \wedge B} \qquad \frac{\Gamma;\vdash A_i}{\Gamma;\vdash A_0 \vee A_1} \qquad \frac{\Gamma;A \vdash B}{\Gamma;\vdash A \supset B}$$

$$\frac{}{\Gamma;\vdash t = t} \qquad \frac{\Gamma;\vdash Pt}{\Gamma;\vdash \exists x.Px} \qquad \frac{\Gamma;\vdash Px}{\Gamma;\vdash \forall x.Px}$$

$$\frac{}{\Gamma,\mu B\vec{t};\vdash \mu B\vec{t}} \qquad \frac{\Gamma;\vdash B(\mu B)\vec{t}}{\Gamma;\vdash \mu B\vec{t}}$$

$$\frac{S \in \mathcal{H} \quad \Gamma;\vdash S\vec{t} \quad ;S\vec{x} \vdash BS\vec{x}}{\Gamma;\vdash \nu B\vec{t}}$$

Fig. 3. Focused proof system for μLJL.

Here, $[S]$ stands for $\lambda\vec{x}.\ [S\vec{x}]$, and the validity of the translation relies on the fact that $\overline{[B]}[S]^\perp\vec{x}$ is the same as $[BS\vec{x}]^\perp$. Note that BS belongs to $\mathcal{H}$ whenever both S and B are in $\mathcal{H}$, meaning that for any p and $\vec{x}$, $Bp\vec{x} \in \mathcal{H}$. The coinduction rule is treated symmetrically, except that in this case B can be in $\mathcal{G}$.

$$\frac{\Gamma \vdash S\vec{t} \quad S\vec{x} \vdash BS\vec{x}}{\Gamma \vdash \nu B\vec{t}} \qquad \longleftrightarrow \qquad \frac{\vdash [\Gamma]^\perp, [S]\vec{t} \quad \vdash [S]^\perp\vec{x}, [B][S]\vec{x}}{\vdash [\Gamma]^\perp, \nu[B]\vec{t}}$$

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 the negative μMALL formulas corresponding to encodings of $\mathcal{H}$ formulas. □

Linear logic provides an appealing proof theoretic setting because of its emphasis on dualities and of its clear separation of concepts (additive vs. multiplicative, asynchronous vs. synchronous). Our experience is that μMALL is a good place to study focusing in the presence of least and greatest fixed point connectives. To get similar results for μLJ, one can either work from scratch entirely within the intuitionistic framework 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 focused proof-system for intuitionistic logic.

$$\begin{array}{ccc} \vdash F & \longrightarrow & \vdash [F] \\ \downarrow & & \downarrow \\ \vdash\Uparrow F & \longleftarrow & \vdash\Uparrow [F] \end{array}$$

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 [Danos et al. 1993; Liang and Miller 2007] in order to minimize the role played by the exponentials in such encodings. Here, as a result of Proposition 4.17, it is possible to get a complete focused system for μLJL that inherits exactly the strong structure of linear μ-focused derivations.

This system is presented in Figure 3. Its sequents have the form $\Gamma; \Gamma' \vdash P$ where Γ' is a multiset of synchronous formulas (fragment $\mathcal{H}$) and the set Γ contains frozen least fixed points in $\mathcal{H}$. First, notice that accordingly with the absence of exponentials in the encoding into linear logic, there is no structural rule. The asynchronous phase takes place on sequents where Γ' is not empty. The synchronous phase processes sequents of the form $\Gamma; \vdash P$, where the focus is without any ambiguity on P. It is impossible to introduce any connective on the right when Γ' is not empty. As will be visible in the following proof of completeness, the synchronous phase in μLJL does not correspond exactly to a synchronous phase in μMALL: it contains rules that are translated into asynchronous μMALL rules, namely implication, universal quantification and coinduction. We introduced this simplification in order to simplify the presentation, which is harmless since there is no choice in refocusing afterwards.

PROPOSITION 4.18 SOUNDNESS AND COMPLETENESS. *The focused proof system for μLJL is sound and complete with respect to μLJL: any focused μLJL derivation of $\Gamma'; \Gamma \vdash P$ can be transformed into a μLJL derivation of $\Gamma', \Gamma \vdash P$; any quasi-finite μLJL derivation of $\Gamma \vdash P$ can be transformed into a μLJL derivation of $\cdot; \Gamma \vdash P$.*

PROOF. The soundness part is trivial: unfocused μLJL derivations can be obtained from focused derivations by removing focusing annotations. Completeness is established using the translation to linear logic as outlined above. Given a μLJL derivation of $\Gamma \vdash P$, we obtain a μMALL derivation of $[\Gamma] \vdash [P]$ using Proposition 4.17. This derivation inherits quasi-finiteness, so we can obtain a μ-focused μMALL derivation of $\vdash \Uparrow [\Gamma]^\perp, [P]$. All sequents of this derivation correspond to encodings of μLJL sequents, always containing a formula that corresponds to the right-hand side of μLJL sequents. By permutability of asynchronous rules, we can require that asynchronous rules are applied on right-hand side formulas only after any other asynchronous rule in our μ-focused derivation. Finally, we translate that focused derivation into a focused μLJL derivation. Let Γ be a multiset of least fixed points in $\mathcal{H}$, Γ' be a multiset of $\mathcal{H}$ formulas, and P be a formula in $\mathcal{G}$.

(1) If there is a μ-focused derivation of $\vdash ([\Gamma]^\perp)^* \Uparrow [\Gamma']^\perp, [P]$ or $\vdash ([\Gamma]^\perp)^*, [P] \Uparrow [\Gamma']^\perp$ then there is a focused μLJL derivation of $\Gamma; \Gamma' \vdash P$.
(2) If there is a μ-focused derivation of $\vdash ([\Gamma]^\perp)^* \Downarrow [P]$ then there is a focused μLJL derivation of $\Gamma; \vdash P$.

We proceed by a simultaneous induction on the μ-focused derivation.

(1) Since $[P]$ is the only formula that may be synchronous, the μ-focused derivation can only start with two switching rules: either $[P]$ is moved to the left of the arrow, in which case we conclude by induction hypothesis (1), or Γ' is empty and $[P]$ is focused on, in which case we conclude by induction hypothesis (2).
If the μ-focused derivation starts with a logical rule, we translate it into a μLJL focused rule before concluding by induction hypothesis. For instance, the & or $\neq$ rule, which can only be applied to a formula in $[\Gamma']^\perp$, respectively correspond to a left disjunction or equality rule. Other asynchronous μMALL rules translate differently depending on whether they are applied on $[\Gamma]^\perp$ or $[P]$: $\invamp$ can correspond to left conjunction or right implication; ν to left μ (induction) or right ν (coinduction); $\forall$ to left $\exists$ or right $\forall$. Note that in the case where $[P]$ is principal, the constraint

on the order of asynchronous rules means that Γ is empty, which is required by synchronous μLJL rule. Finally, freezing is translated by the μLJL rule moving a least fixed point from Γ' to Γ.

(2) If the μ-focused derivation starts with the switching rule releasing focus from $[P]$ we conclude by induction hypothesis (1). Otherwise it is straightforward to translate the first rule and conclude by induction hypothesis (2): $\otimes$, $\oplus$, $=$, $\exists$ and μ respectively map to the right rules for $\wedge$, $\vee$, $=$, $\exists$ and μ.

Note, however, that the tensor rule splits frozen formulas in $([\Gamma]^{\perp})^*$, while the right conjunction rule of μLJL does not. This is harmless because weakening is obviously admissible for the frozen context of μLJL focused derivations. This slight mismatch means that we would still have a complete focused proof system for μLJL if we enforced a linear use of the frozen context. We chose to relax this constraint, as it does not make a better system for proof search. $\qquad\square$

Although μLJL is only a small fragment of 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, of the form $\forall x.\ P\,x \supset Q\,x$, are in $\mathcal{G}$ provided that P and Q are in $\mathcal{H}$. Further, implications can be chained through a greatest fixed point construction, which allows to specify various relations on process behaviors. For example, provided that one-step transitions $u \to v$ are specified in $\mathcal{H}$, simulation is naturally expressed in $\mathcal{G}$ as follows.

$$\nu S \lambda x \lambda y.\ \forall x'.\ x \to x' \supset \exists y'.\ y \to y' \wedge S\,x'\,y'$$

Finally, the theorems about natural numbers presented in Section 2.6 are also in $\mathcal{G}$. Although a formula in $\mathcal{G}$ can *a priori* be a theorem in μLJ but not in μLJL, we have shown [Baelde 2009] that μLJL is complete for inclusions of nondeterministic finite automata — $\mathcal{A} \subseteq \mathcal{B}$ being expressed naturally as $\forall w.\ [\mathcal{A}]w \supset [\mathcal{B}]w$.

Interestingly, the μLJL fragment has already been identified in LINC [Tiu et al. 2005] and the Bedwyr system [Baelde et al. 2007] implements a proof-search strategy for it that is complete for finite behaviors, that is, proofs without (co)induction nor axiom rules, where a fixed point has to be treated in a finite number of unfoldings. This strategy coincides with the focused system for μLJL, where the finite behavior restriction corresponds to dropping the freezing rule, obtaining a system where proof search consists in eagerly eliminating any left-hand side (asynchronous) formula before working on the goal (right-hand side), without ever performing any contraction or weakening. The logic μLJ is closely related to LINC, the main difference being the generic quantifier ∇, which allows to specify and reason about systems involving variable binding, such as the π-calculus [Tiu 2005]. But we have shown [Baelde 2008b] that ∇ can be added in an orthogonal fashion in μLJ (or μMALL) without affecting focusing results.

5. CONCLUSION

We have defined μMALL, a minimal and well-structured proof system featuring fixed points, and established the two main properties for that logic. The proof of cut elimination is the first contribution of this paper, improving on earlier work and contributing to the understanding of related works. The second and main contribution is the study and design of focusing for that logic. This challenging extension of focused proofs forces us to reflect on the foundations of focusing, and brought new proof search applications of focusing. We have shown that μMALL is a good logic for the foundational study of fixed points, but also a rich system that can directly support interesting applications:

combining observations on admissible structural rules with our μ-focused system, we were able to derive a focused proof system for an interesting fragment of μLJ.

Although carried out in the simple logic μMALL, this work on fixed points has proved meaningful in richer logics. We have extended our focusing results to μLL and μLJ [Baelde 2008a], naturally adapting the designs and proof techniques developed in this article. However, focused systems obtained by translating the target logic into μMALL (or μLL) are often not fully satisfying, and better systems can be crafted and proved complete from scratch, using the same techniques as for μMALL, with a stronger form of balancing that imposes uniform asynchronous choices over all contractions of a formula.

Further work includes various projects relying on μMALL and its extensions, from theory to implementation. But we shall focus here on important open questions that are of general interest concerning this formalism. An obvious first goal would be to strengthen our weak normalization proof into a strong normalization result. The relationship between cut elimination and focusing also has to be explored more; we conjectured that focusing preserves the identity (cut elimination behavior) of proofs, and that the notion of quasi-finiteness could be refined so as to be preserved by cut elimination. Finally, it would be useful to be able to characterize and control the complexity of normalization, and consequently the expressiveness of the logic; here, one could explore different classes of (co)invariants, or other formulations of (co)induction.

ACKNOWLEDGMENTS

REFERENCES

ALVES, S., FERNÁNDEZ, M., FLORIDO, M., AND MACKIE, I. 2006. The power of linear functions. In *Proceedings of the International Workshop on Computer Science Logic*. 119–134.

ANDREOLI, J.-M. 1992. Logic programming with focusing proofs in linear logic. *J. Logic Comput. 2*, 3, 297–347.

ANDREOLI, J. M. AND PARESCHI, R. 1991. Linear objects: Logical processes with built-in inheritance. *New Gen. Comput. 9*, 3–4, 445–473.

APT, K. R. AND VAN EMDEN, M. H. 1982. Contributions to the theory of logic programming. *J. ACM 29*, 3, 841–862.

BAELDE, D. 2008a. A linear approach to the proof-theory of least and greatest fixed points. Ph.D. thesis, Ecole Polytechnique.

BAELDE, D. 2008b. On the expressivity of minimal generic quantification. In *Proceedings of the International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice*. A. Abel and C. Urban Eds., Electronic Notes in Theoretical Computer Science, No. 228, 3–19.

BAELDE, D. 2009. On the proof theory of regular fixed points. In *Proceedings of the 18th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*. M. Giese and A. Waller Eds., Lecture Notes in Computer Science, vol. 5607, 93–107.

BAELDE, D. AND MILLER, D. 2007. Least and greatest fixed points in linear logic. In *Proceedings of the International Conference on Logic for Programming and Automated Reasoning (LPAR)*. N. Dershowitz and A. Voronkov Eds., Lecture Notes in Computer Science, vol. 4790, 92–106.

BAELDE, D., GACEK, A., MILLER, D., NADATHUR, G., AND TIU, A. 2007. The Bedwyr system for model checking over syntactic expressions. In *Proceedings of the 21th Conference on Automated Deduction (CADE)*. F. Pfenning Ed., Lecture Notes in Computer Science, vol. 4603, Springer, 391–397.

BAELDE, D., MILLER, D., AND SNOW, Z. 2010. Focused inductive theorem proving. In *Proceedings of the 5th International Joint Conference on Automated Reasoning*. J. Giesl and R. Hahnle Eds., Lecture Notes in Computer Science, vol. 6173, 278–292.

BARENDREGT, H. 1992. Lambda calculus with types. In *Handbook of Logic in Computer Science*. S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum Eds., Vol. 2. Oxford University Press, 117–309.

BROTHERSTON, J. 2005. Cyclic proofs for first-order logic with inductive definitions. In *Proceedings of the 18th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*. B. Beckert Ed., Lecture Notes in Computer Science, vol. 3702, Springer, 78–92.

BURRONI, A. 1986. Récursivité graphique (lère partie): Categorie des fonctions recursives primitives formelles. *Cah. Topologie Geom. Differ. Categoriques 27*, 1, 49–79.

CHAUDHURI, K. AND PFENNING, F. 2005. Focusing the inverse method for linear logic. In *Proceedings of the International Workshop on Computer Science Logic*. C.-H. L. Ong Ed., Lecture Notes in Computer Science, vol. 3634, Springer, 200–215.

CHAUDHURI, K., PFENNING, F., AND PRICE, G. 2008. A logical characterization of forward and backward chaining in the inverse method. *J. Automat. Reason. 40*, 2–3, 133–177.

CLAIRAMBAULT, P. 2009. Least and greatest fixpoints in game semantics. In *Proceedings of the 12th International Conference on the Foundations of Software Science and Computational Structures*. L. de Alfaro Ed., Lecture Notes in Computer Science, vol. 5504, Springer, 16–31.

DANOS, V., JOINET, J.-B., AND SCHELLINX, H. 1993. The structure of exponentials: Uncovering the dynamics of linear logic proofs. In *Proceedings of the Kurt Gödel Colloquium*. G. Gottlob, A. Leitsch, and D. Mundici Eds., Lecture Notes in Computer Science, vol. 713, Springer, 159–171.

DANOS, V., JOINET, J.-B., AND SCHELLINX, H. 1995. LKT and LKQ: Sequent calculi for second order logic based upon dual linear decompositions of classical implication. In *Advances in Linear Logic*. J.-Y. Girard, Y. Lafont, and L. Regnier Eds., London Mathematical Society Lecture Note Series, vol. 222, Cambridge University Press, 211–224.

DELANDE, O. AND MILLER, D. 2008. A neutral approach to proof and refutation in MALL. In *Proceedings of the 23rd Symposium on Logic in Computer Science*. F. Pfenning Ed., IEEE, 498–508.

DELANDE, O., MILLER, D., AND SAURIN, A. 2010. Proof and refutation in MALL as a game. *Ann. Pure Appl. Logic 161*, 5, 654–672.

GIRARD, J.-Y. 1987. Linear logic. *Theor. Comput. Sci. 50*, 1–102.

GIRARD, J.-Y. 1992. A fixpoint theorem in linear logic. An email posting to the mailing list linear@cs.stanford.edu.

GIRARD, J.-Y. 2001. Locus solum: From the rules of logic to the logic of rules. *Math. Struct. Comput. Sci. 11*, 3, 301–506.

HODAS, J. AND MILLER, D. 1994. Logic programming in a fragment of intuitionistic linear logic. *Inform. Comput. 110*, 2, 327–365.

HUET, G. 1975. A unification algorithm for typed A-calculus. *Theor. Comput. Sci. 1*, 27–57.

LAURENT, O. 2002. Etude de la polarisation en logique. Ph.D. thesis, Université Aix-Marseille II.

LAURENT, O. 2004. A proof of the focalization property of linear logic. Unpublished note.

LAURENT, O., QUATRINI, M., AND DE FALCO, L. T. 2005. Polarized and focalized linear and classical proofs. *Ann. Pure Appl. Logic 134*, 2–3, 217–264.

LIANG, C. AND MILLER, D. 2007. Focusing and polarization in intuitionistic logic. In *Proceedings of the International Workshop on Computer Science Logic*. J. Duparc and T. A. Henzinger Eds., Lecture Notes in Computer Science, vol. 4646, Springer, 451–465.

MATTHES, R. 1999. Monotone fixed-point types and strong normalization. In *Proceedings of the International Workshop on Computer Science Logic*. G. Gottlob, E. Grandjean, and K. Seyr Eds., Lecture Notes in Computer Science, vol. 1584, 298–312.

MCDOWELL, R. AND MILLER, D. 2000. Cut-elimination for a logic with definitions and induction. *Theor. Comput. Sci. 232*, 91–119.

MENDLER, N. P. 1991. Inductive types and type constraints in the second order lambda calculus. *Ann. Pure Appl. Logic 51*, 1, 159–172.

MILLER, D. 1992. Unification under a mixed prefix. *J. Symb. Comput. 14*, 4, 321–358.

MILLER, D. 1996. Forum: A multiple-conclusion specification logic. *Theor. Comput. Sci. 165*, 1, 201–232.

MILLER, D. AND NIGAM, V. 2007. Incorporating tables into proofs. In *Proceedings of the International Workshop on Computer Science Logic*. J. Duparc and T. A. Henzinger Eds., Lecture Notes in Computer Science, vol. 4646, Springer, 466–480.

MILLER, D. AND PIMENTEL, E. 2010. A formal framework for specifying sequent calculus proof systems. Available from authors' Web sites.

MILLER, D. AND SAURIN, A. 2006. A game semantics for proof search: Preliminary results. In *Proceedings of the Mathematical Foundations of Programming Semantics (MFPS '05)*. Electronic Notes in Theoretical Computer Science, vol. 155, 543–563.

MILLER, D. AND SAURIN, A. 2007. From proofs to focused proofs: A modular proof of focalization in linear logic. In *Proceedings of the International Workshop on Computer Science Logic*. J. Duparc and T. A. Henzinger Eds., Lecture Notes in Computer Science, vol. 4646, Springer, 405–419.

MILLER, D. AND TIU, A. 2005. A proof theory for generic judgments. *ACM Trans. Comput. Logic 6*, 4, 749–783.

MILLER, D., NADATHUR, G., PFENNING, F., AND SCEDROV, A. 1991. Uniform proofs as a foundation for logic programming. *Ann. Pure Appl. Logic 51*, 125–157.

MOMIGLIANO, A. AND TIU, A. 2003. Induction and co-induction in sequent calculus. In *Proceedings of the International Workshop on Types for Proofs and Programs*. M. Coppo, S. Berardi, and F. Damiani Eds., Lecture Notes in Computer Science, vol. 3085, 293–308.

NIGAM, V. 2009. Exploiting non-canonicity in the sequent calculus. Ph.D. thesis, Ecole Polytechnique.

SANTOCANALE, L. 2001. A calculus of circular proofs and its categorical semantics. BRICS Report Series RS-01-15, BRICS, Department of Computer Science, University of Aarhus.

SCHROEDER-HEISTER, P. 1993. Rules of definitional reflection. In *Proceedings of the 8th Annual Symposium on Logic in Computer Science*. M. Vardi, Ed., IEEE, 222–232.

TIU, A. 2004. A logical framework for reasoning about logical specifications. Ph.D. thesis, Pennsylvania State University.

TIU, A. 2005. Model checking for π-calculus using proof search. In *Proceedings of the International Conference on Concurrency Theory*. M. Abadi and L. de Alfaro Eds., Lecture Notes in Computer Science, vol. 3653, Springer, 36–50.

TIU, A. AND MOMIGLIANO, A. 2010. Cut elimination for a logic with induction and co-induction. CoRR abs/1009.6171.

TIU, A., NADATHUR, G., AND MILLER, D. 2005. Mixing finite success and finite failure in an automated prover. In *Proceedings of the Workshop on Empirically Successful Automated Reasoning in Higher-Order Logics (ESHOL'05)*. 79–98.