Equality and fixpoints in the calculus of structures

Kaustuv Chaudhuri, Nicolas Guenot

Equality and Fixpoints in the Calculus of Structures

Kaustuv Chaudhuri
INRIA, France
kaustuv.chaudhuri@inria.fr

Nicolas Guenot
IT University of Copenhagen, Denmark
ngue@itu.dk

December 5, 2014

Abstract

The standard proof theory for logics with equality and fixpoints suffers from limitations of the sequent calculus, where reasoning is separated from computational tasks such as unification or rewriting. We propose in this paper an extension of the calculus of structures, a deep inference formalism, that supports incremental and contextual reasoning with equality and fixpoints in the setting of linear logic. This system allows deductive and computational steps to mix freely in a continuum which integrates smoothly into the usual versatile rules of multiplicative-additive linear logic in deep inference.

1 Introduction

It is a common design pattern for computational logic proof systems to compartmentalize and separate the computational aspects from the rest of the system. Examples include μMALL [2] and LINC [19] that separate unification, Deduction Modulo [9] and Super Deduction [4] that separate term rewriting, and any proof system with constraints that separates algebraic manipulations of constraint terms. The record of a formal proof in these systems omits the computational traces, which are reconstructed as needed by the consumers of the proof, usually computerized proof checkers. While this design is fairly successful in automated certifying proof search, it has a number of undesirable features from the perspective of proof representation. First, because the nature of the omitted computations differs between systems, implementations of different systems cannot generally check each others' proofs – at least, not without reimplementing the computational kernels. Second, such proofs tend to restrict computations to rigid positions, such as the leaves, which limits the possibilities for reorganizing the proofs. Third, the computations tend to have a separate language that makes them opaque to ordinary logical manipulation. Finally, the computational phases are nearly always atomic, so it is inordinately difficult, if at all possible, to perform the computations incrementally, partially, or in parallel, or to substitute one computation inside another.

In this paper, we revisit the question of combining computation and deduction in proof systems, but with a change of foundational perspective. Instead of adding computational features to a deductive system, we start with a computational system from the outset that can perform deduction. Specifically, we use the *calculus of structures* (COS), which is a *deep inference* system that blurs the distinction between formula and sequent (or, equivalently, between formula and judgement). A COS proof is organized as a linear rewrite sequence from the desired formula to logical truth. Each step corresponds to the embedding of a valid entailment of the logic into a *formula context*, so it is just as natural to substitute a COS proof into another proof with a hole as it is to substitute a formula into a formula context. Moreover, the design of the COS does not preclude the use of shallow proofs. Indeed, not only can the COS simulate sequent proofs *at any depth*, but it also has deep manifestations of such proof-theoretic phenomena as cut-elimination and focusing, familiar from the sequent calculus [17, 8].

The most striking feature of the COS is that its inference rules are largely *incremental*, meaning that they can be applied[1] by performing a fixed and constant amount of work. To illustrate: multiplicative rules are implemented in the COS not by a global division of all the hypotheses into disjoint branches of the proof, but rather by a sequence of smaller routing steps where individual subformulas are sent to components of the multiplicative connective. This incremental nature gives the COS much more permutative freedom than in shallow systems such as the sequent calculus or natural deduction. It is natural in the COS, for instance, to share common subproofs across additive branches by performing the common proof before the additive split; this can only be achieved in shallow systems indirectly by means of cuts or extensions. The COS also supports natural metaphors for proof construction by *direct manipulation* that are not possible to express in shallow systems [7].

[1] In this paper, we will read inference rules from conclusion to premise(s).

Unfortunately, despite these desirable properties, the literature on the COS has some inexplicably wide gaps that has heretofore limited its use as a proof formalism for computational logic. Nearly the entirety of the published work on the COS deals with propositional logics, which are of no use for reasoning about data structures—particularly recursive data structures—that are the *sine qua non* of computation. The main contribution of this work is an extension of the COS to first-order classical multiplicative additive linear logic (MALL) with support for equality and fixpoint predicates, which raises the expressivity of the COS to the level of μMALL [2] for first-order terms. Our treatment of these extensions stays as close to the spirit of the COS as possible; the rules for equality and fixponts are incremental and contextual, and can be freely interleaved and permuted with other inference rules. Indeed, we show that equality is a critical ingredient for an incremental treatment of the first-order predicates.

Supporting equality requires the addition of rules for decomposing equations on larger terms to those on smaller terms. For purely positive occurrences of equations, this amounts to the rules of a *unification logic* that essentially implement Robinson's unification algorithm. Symmetry and transitivity are freely admissible in this fragment. The *atomic identity* rule, which requires two $\parr$-linked dual atomic formulas, can be replaced with an incremental version that only requires dual predicate symbols; the term arguments are checked point-wise for equality. The situation becomes more complex in the presence of disequations—negative occurrences of equations—because symmetry and transitivity become essential rules. Transitivity is particularly problematic as it requires the invention of an interpolant that is not structurally present in the conclusion. Fortunately, we can leverage depth here to effectively eliminate this problem. The essential idea is that we can view transitivity not as $(s = t) \otimes (t = u) \vdash (s = u)$ but as $(t = u) \vdash (s = t) \multimap (s = u)$, which are logically equivalent, but the latter form contains all the terms on the right hand side, which forms the conclusion of the COS rule. All the inference rules become analytic on terms.

The most important use of equations in computational logic is in the theory of uninterpreted functions, where all term constants are viewed as *constructors*, *i.e.*, they are injective and discriminating. We distinguish this equality from the mathematical notion by calling it *intensional*. Intensional disequations are very powerful: they are interpreted as an enumeration over all possible ways to prove the converse, which is the essence of reasoning by cases. In the μMALL sequent calculus [2], this rule is written using the *complete set of unifiers* (csu) as follows:

$$\frac{\text{for every } \theta \in \text{csu}(s,t), \vdash \theta\Gamma}{\vdash \Gamma, (s \neq t)}$$

This rule has two big problems. First, as said before, the csu computation is not part of the proof system, so any proof consumer must be able to recompute the csu. This requirement is worse than it appears: when exchanging formal proof objects, the proof producer and consumer must agree not only on the contents of the csu but also on their *order*. This is a significant restriction on the independence of the consumer from the producer. Second, this rule makes a copy of the entire context Γ for every element of the csu. This copying is not mere additive sharing, which can often be implemented by shared pointers, because the various copies of the context are instantiated by different substitutions and hence may have significant varying subsets. To make the csu computation more incremental in the COS, we show how to express it using the connectives of MALL that already exist, and we then rely on the incremental nature of the rules for these connectives.

Intensional equality comes fully alive with the addition of least and greatest fixpoints. We show how to adapt the unfolding and coinduction rules of μMALL to an incremental and contextual form. The main challenge is that the coinduction rule of μMALL has the form of an essential cut where one of the premises establishes that the cut-formula is a post-fixpoint. This premise has to be isolated from the rest of the sequent, particularly the linear hypotheses. Such an isolation is not possible in the COS for MALL itself without making the coinduction rule non-contextual. Our solution is to extend MALL with a connective that simply records that its argument is isolated; in effect it is similar to the ! connective of the full linear logic, except it has no dual and cannot be contracted or weakened. The only operations allowed on this connective to delete it if its operand has successfully been rewritten to $\mathsf{1}$.

The addition of fixpoints also adds a new wrinkle to the atomic identity rule, which must now check that the *same* possible large fixpoint expression occurs in dual $\parr$-linked forms. We construct an incremental version of this rule that only checks for inclusion of the least fixpoint in the greatest fixpoint. Surprisingly, this turns out to be sound and complete by means of a fairly simple second-order substitution argument. Moreover, it can be seen as inlining a particular kind of coinduction where the least fixpoint is used as an invariant for the greatest fixpoint, so we re-use the isolation trick we already used to implement the (co-)induction rule.

The paper is structured as follows. In Section 2 we sketch the COS for first-order MALL. Section 3 incorporates equality into this calculus in stages, starting with purely positive occurrences and progressing to the full intensional equality. Section 4 then extends the system with least and greatest fixpoints, and gives a few illustrative examples. Related work and possibilities for future work are summarized in Section 5.

$$\frac{\xi\{1\}}{\xi\{a \parr \overline{a}\}}\,\mathsf{aid} \qquad \frac{\xi\{(A \parr B) \parr C\}}{\xi\{A \parr (B \parr C)\}}\,\mathsf{assoc} \qquad \frac{\xi\{B \parr A\}}{\xi\{A \parr B\}}\,\mathsf{comm}$$

$$\frac{\xi\{(A \parr C) \otimes B\}}{\xi\{(A \otimes B) \parr C\}}\,\mathsf{swl} \qquad \frac{\xi\{B \otimes (A \parr C)\}}{\xi\{A \parr (B \otimes C)\}}\,\mathsf{swr} \qquad \frac{\xi\{1\}}{\xi\{1 \otimes 1\}}\,\mathsf{onem} \qquad \frac{\xi\{A\}}{\xi\{A \oplus B\}}\,\mathsf{chl} \qquad \frac{\xi\{B\}}{\xi\{A \oplus B\}}\,\mathsf{chr} \qquad \frac{\xi\{[t/x]A\}}{\xi\{\exists x.\, A\}}\,\mathsf{wit}$$

$$\frac{\xi\{(A \parr B) \,\&\, (A \parr C)\}}{\xi\{A \parr (B \,\&\, C)\}}\,\mathsf{dist} \qquad \frac{\xi\{\top\}}{\xi\{A \parr \top\}}\,\mathsf{abs} \qquad \frac{\xi\{1\}}{\xi\{\top\}}\,\mathsf{top} \qquad \frac{\xi\{1\}}{\xi\{1 \,\&\, 1\}}\,\mathsf{onea} \qquad \frac{\xi\{A\}}{\xi\{A \parr \bot\}}\,\mathsf{bot} \qquad \frac{\xi\{\forall x.\, (A \parr B)\}}{\xi\{A \parr \forall x.\, B\}}\,\mathsf{all} \qquad \frac{\xi\{1\}}{\xi\{\forall x.\, 1\}}\,\mathsf{vac}$$

Notes:

- In wit, the substitution $[t/x]A$ is capture-avoiding
- In all, x is not free in A

Figure 1: The system *MAS*.

2 *MALL* in the Calculus of Structures

We begin by summarizing the calculus of structures for first-order classical multiplicative-additive linear logic (*MALL*), based on the design of [8]. This logic is selected primarily because it is the basis for the μ*MALL* system, but linearity gives us the additional ability to be precise about the multiplicities of formulas, and requires us to be explicit in our use of contraction and weakening. We omit the exponential connectives of linear logic as they are largely redundant with fixpoints. The terms $(s, t, \dots)$, atomic formulas $(a, b, \dots)$ and arbitrary formulas $(A, B, \dots)$ of our language have the following grammar, where $x, y, \dots$ range over term variables, $\mathsf{f}, \mathsf{g}, \dots$ range over function symbols, and $\mathsf{p}, \mathsf{q}, \dots$ range over predicates.

$$\begin{aligned} s, t, \dots &::= x \mid \mathsf{f}(t_1, \dots, t_n) \\ a, b, \dots &::= \mathsf{p}(t_1, \dots, t_n) \\ A, B, \dots &::= a \mid A \otimes B \mid 1 \mid A \oplus B \mid 0 \mid \exists x.\, A \\ &\quad \mid \overline{a} \mid A \parr B \mid \bot \mid A \,\&\, B \mid \top \mid \forall x.\, A \end{aligned}$$

We assume that variables, function symbols, and predicate symbols are drawn from pairwise disjoint infinite sets, and that the arity of every function and predicate symbols is fixed. We write $\vec{s}$ to stand for a list of terms $s_1, \dots, s_n$ (with $n \geq 0$). By convention, we omit the empty argument list for nullary function and predicate symbols. We write $\overline{A}$ for the negation of A, defined as usual in terms of De Morgan duals. Formulas are in negation-normal form, so the only occurrence of negation in the syntax is in negated atoms, indicated with an overline. We write $A \multimap B$ as usual to stand for $\overline{A} \parr B$, $A \vdash B$ to assert that the formula $A \multimap B$ is true in *MALL*, and $A \dashv\vdash B$ to assert that $A \vdash B$ and $B \vdash A$.

The main feature of the calculus of structures (*COS*) is that the rules are allowed to be applied in any *formula context*, which is a formula with a single occurrence of a *hole* ($\Box$). The grammar of formula contexts $(\xi, \rho, \dots)$ is thus:

$$\xi, \rho, \dots ::= \Box \mid A \star \xi \mid \xi \star A \mid Qx.\, \xi$$

where $\star \in \{\otimes, \oplus, \parr, \&\}$ and $Q \in \{\exists, \forall\}$. We write $\xi\{A\}$ for the replacement of the single hole in ξ with A, called a *substitution*, which is allowed to capture the free variables in A; we call A the *selected formula* in the notation $\xi\{A\}$. Substitutions are used to turn shallow entailments and admissible rules into their deep variants.

Theorem 1. *For any A, B, and ξ,*

- *if $A \vdash B$, then also $\xi\{A\} \vdash \xi\{B\}$; and*
- *if the truth of A implies the truth of B, then the truth of $\xi\{A\}$ implies the truth of $\xi\{B\}$.*

Proof. By induction on the structure of ξ. □

The inference rules for the system *MAS* for this logic are defined in terms of substitutions. The full system is shown in Figure 1. This system is a minor variant of the system *LS* from [8]. Each rule contains exactly one premise and conclusion and corresponds to a valid entailment of linear logic. All the rules except assoc and comm are oriented in such a fashion that the premise is always simpler than the conclusion in the following sense: either the selected formula in the premise is a strict subformula of that of the conclusion, or one of the connectives &, $\otimes$, or $\forall$ has a larger scope in the premise than in the conclusion, or a selected 1 in the premise is $\top$ in the conclusion. We will adopt the convention of omitting mentions of the assoc and comm rules, but use them implicitly.

Definition 2 (standard notions)**.** Let $\mathcal{S}$ be any unary formula inference system, *i.e.*, a system where the sole premise and conclusion of each rule schema are formula schemas.

- An $\mathcal{S}$-*derivation* ϕ of formula B *from* formula A, written $\phi : A \xrightarrow{\mathcal{S}} B$, is a chained sequence of instances of the rules of $\mathcal{S}$ such that the topmost rule instance has premise A and the bottom-most rule instance has conclusion B.
- An $\mathcal{S}$-*proof* π *of* A is a derivation $\pi : 1 \xrightarrow{\mathcal{S}} A$.
- The formula B is $\mathcal{S}$-*derivable from* A, written as $A \xrightarrow{\mathcal{S}} B$, if there exists an $\mathcal{S}$-derivation of B from A. The formula A is $\mathcal{S}$-*provable* if there exists an $\mathcal{S}$-proof of A.
- A schematic rule is $\mathcal{S}$-*admissible* if in every instance of the schema whenever the premise is $\mathcal{S}$-provable then so is the conclusion.

When $\mathcal{S}$ is understood, the prefix "$\mathcal{S}$-" will be elided.

Theorem 3 (Soundness of MAS)**.** *For any formulas A and B, if $A \xrightarrow{MAS} B$, then $A \vdash B$.*

Proof. Each inference rule of MAS is a valid entailment. The result follows from the composability of entailment for MALL, *i.e.*, that $A \vdash B$ and $B \vdash C$ imply that $A \vdash C$. □

Theorem 4 (Completeness of MAS)**.** *If A is a true formula of MALL, then $1 \xrightarrow{MAS} A$.*

Sketch. If A is true, then there must be a derivation of $\vdash A$ in a cut-free sequent calculus for MALL, such as the one shown in Figure 4. It is immediate by inspection that each inference rule of this calculus can be simulated in MAS (for more detail on this correspondence, see [8]). The occurrences of 1 corresponding to the branches of the sequent derivation are then combined with onem or onea. □

It should be noted that MAS has a natural dual system where every inference rule is inverted and the premise and conclusion are dualized. The combination of MAS and its dual system is a purely symmetric system (a variant of SLS [17]), which also includes the dual of the aid rule,

$$\frac{\xi\{a \otimes \overline{a}\}}{\xi\{\bot\}}\,\mathsf{acut}$$

General cuts in SLS can be reduced to these atomic cuts, which can then be shown to be admissible in MAS. Much of this technique has been worked out for the full propositional linear logic in [17, 8]. We expect that the extension of such results to the first order case would be straightforward.

The system MAS also admits a number of *congruence* rules that are standard in the literature on the COS, such as in [17]. These rules are left out of MAS, but we will use them in the remainder of this paper if needed to simplify the presentation.

$$\frac{\xi\{B * A\}}{\xi\{A * B\}} \quad \frac{\xi\{(A * B) * C\}}{\xi\{A * (B * C)\}} \quad \frac{\xi\{A\}}{\xi\{A * \ddagger\}}$$

Here, $(*, \ddagger) \in \{(\otimes, 1), (\oplus, 0), (\&, \top)\}$. We will also consider a derivation to be a proof if the premise can be derived from the simple rules on logical constants: $\{\mathsf{onea}, \mathsf{onem}, \mathsf{top}, \mathsf{vac}\}$.

3 Equality

3.1 Positive Equality and Incremental Unification

Let us turn to an ignored corner of MAS (Figure 1): the aid rule. This rule requires the *same* atom to be present in dual forms across the $\parr$. This requirement forces instantiations (the wit rule) to be performed before (*i.e.*, below) the aid rule, which makes the calculus rigid with respect to instantiations. It also requires traversing arbitrarily large structures to check identity, which is against the incrementality principle. To make the rule incremental and more flexible, we replace it with the following variant.

$$\frac{\xi\{(\vec{s} = \vec{t})\}}{\xi\left\{\mathsf{p}(\vec{s}) \parr \overline{\mathsf{p}(\vec{t})}\right\}}\,\mathsf{match}$$

Here, $=$ is a binary predicate symbol (written infix, as usual) denoting that its two arguments are equal terms. In this rule and in the rest of this paper, we will use the convention that if $\vec{s} = s_1, \ldots, s_n$ and $\vec{t} = t_1, \ldots, t_n$, then the formula $(\vec{s} = \vec{t})$ stands for $(s_1 = t_1) \otimes \cdots \otimes (s_n = t_n)$ if $n \geq 1$ and for 1 if $n = 0$. Thus, the requirement

that the two dual atoms be exactly identical in the aid is relaxed to produce a collection of residual equality obligations for the term arguments.

To reason about such positive equalities, we need just the following rules of reflexivity (refl) and congruence (cong):

$$\frac{\xi\{1\}}{\xi\{x = x\}}\,\mathsf{refl} \qquad \frac{\xi\{(\vec{s} = \vec{t})\}}{\xi\{\mathsf{f}(\vec{s}) = \mathsf{f}(\vec{t})\}}\,\mathsf{cong}$$

Observe that the refl rule is limited to variables. This pair of rules suffices to prove $s = s$ for any term s. Of course, these rules may now be interleaved with other logical rules, including instantiation.

An important side-effect of relaxing aid into match is that existential instantiation can fruitfully be made lazy as well by adding the following rule:

$$\frac{\xi\{\exists x.\,(A \parr B)\}}{\xi\{A \parr \exists x.\,B\}}\,\mathsf{ex}$$

where, as with all, we require that x is not free in A. This rule is easily seen to be admissible in *MAS*, but it only becomes useful when aid is not forced to end a branch.

Definition 5 (System *MAUS*)**.** Let *MAUS* stand for

$$(\mathit{MAS} \setminus \{\mathsf{aid}\}) \cup \{\mathsf{match}, \mathsf{refl}, \mathsf{cong}, \mathsf{ex}\}$$

Let us finish this short section on positive equalities by noting a few interesting features.

Example 6. *In MAUS there is a continuum between logic and unification. Consider this MAUS derivation:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\exists x.\,\forall y.\,\exists z.\,\big((\mathsf{f}(x) = \mathsf{f}(\mathsf{f}(c))) \otimes (y = z)\big)}{\exists x.\,\forall y.\,\exists z.\,\big(\mathsf{p}(\mathsf{f}(x), y) \multimap \mathsf{p}(\mathsf{f}(\mathsf{f}(c)), z)\big)}\,\mathsf{match}}{\exists x.\,\forall y.\,\big(\mathsf{p}(\mathsf{f}(x), y) \multimap \exists z.\,\mathsf{p}(\mathsf{f}(\mathsf{f}(c)), z)\big)}\,\mathsf{ex}}{\exists x.\,\big((\exists y.\,\mathsf{p}(\mathsf{f}(x), y)) \multimap \exists z.\,\mathsf{p}(\mathsf{f}(\mathsf{f}(c)), z)\big)}\,\mathsf{all}}{\big(\forall x.\,\exists y.\,\mathsf{p}(\mathsf{f}(x), y)\big) \multimap \exists z.\,\mathsf{p}(\mathsf{f}(\mathsf{f}(c)), z)}\,\mathsf{ex}$$

Instantiations of the existential variables x and z are not forced before the application of match*. The goal formula is reduced to a* unification problem *consisting only of the quantifiers and (conjunctions of) equality. There is neither a syntactic separation between the two kinds of problems, nor are deductive and unification phases necessarily separate. Unification can be done incrementally using the rules* {refl, cong, wit}*:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall y.\,\big(1 \otimes 1\big)}{\forall y.\,\big(1 \otimes (y = y)\big)}\,\mathsf{refl}}{\forall y.\,\exists z.\,\big(1 \otimes (y = z)\big)}\,\mathsf{wit}}{\forall y.\,\exists z.\,\big(c = c \otimes (y = z)\big)}\,\mathsf{refl}}{\forall y.\,\exists z.\,\big((\mathsf{f}(c) = \mathsf{f}(c)) \otimes (y = z)\big)}\,\mathsf{cong}}{\exists x.\,\forall y.\,\exists z.\,\big((x = \mathsf{f}(c)) \otimes (y = z)\big)}\,\mathsf{wit}}{\exists x.\,\forall y.\,\exists z.\,\big((\mathsf{f}(x) = \mathsf{f}(\mathsf{f}(c))) \otimes (y = z)\big)}\,\mathsf{cong}$$

Example 7. *Since MAUS derivations can postpone applications of the* wit *rule, it is important to ensure that it still respects the usual eigenvariable condition for quantifiers. Consider the following example of an invalid quantifier switch.*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\exists x.\,\forall y.\,\exists z.\,\forall w.\,(x = w) \otimes (y = z)}{\exists x.\,\forall y.\,\exists z.\,\forall w.\,\big(\mathsf{p}(x, y) \multimap \mathsf{p}(w, z)\big)}\,\mathsf{match}}{\exists x.\,\forall y.\,\exists z.\,\big(\mathsf{p}(x, y) \multimap (\forall x'.\,\mathsf{p}(x', z))\big)}\,\mathsf{all}}{\exists x.\,\forall y.\,\big(\mathsf{p}(x, y) \multimap (\exists y'.\,\forall x'.\,\mathsf{p}(x', y'))\big)}\,\mathsf{ex}}{\exists x.\,\big((\exists y.\,\mathsf{p}(x, y)) \multimap (\exists y.\,\forall x'.\,\mathsf{p}(x', y))\big)}\,\mathsf{all}}{(\forall x.\,\exists y.\,\mathsf{p}(x, y)) \multimap (\exists y.\,\forall x.\,\mathsf{p}(x, y))}\,\mathsf{ex}$$

The resulting unification problem is not solvable, because there is no capture-avoiding substitution for x that will make $\forall w.\,x = w$ true.

Example 8 (Occurs Check)**.** *As* MAUS *has neither a rule of transitivity nor a means of introducing new* $\exists$*-quantified variables, it follows that there are only a finite number of simplification steps (*refl *and* cong*) that can be applied to a given unification problem. In other words,* MAUS *can only establish the equality of finite terms. This can be seen a variant of the* occurs check *of Robinson's unification algorithm. To illustrate, the formula* $\exists x.\, \mathsf{p}(x) \multimap \mathsf{p}(\mathsf{f}(x))$*, which has no finite witnesses, has the (incomplete)* MAUS *derivation:*

$$\dfrac{\dfrac{t = \mathsf{f}(t)}{\exists x.\, x = \mathsf{f}(x)}\,\mathsf{wit}}{\exists x.\, \mathsf{p}(x) \multimap \mathsf{p}(\mathsf{f}(x))}\,\mathsf{mat}$$

The only way to proceed with that instance of wit *is to exhibit a term* t *for which* $t = \mathsf{f}(t)$*.*

The previous example indicates that one can often fold the occurs check for positive equality directly into wit as follows, where $\mathrm{vars}(t) \cap (\{x\} \cup \mathrm{bvs}(\xi')) = \emptyset$.[2]

$$\dfrac{\xi\{[t/x]\xi'\{1\}\}}{\xi\{\exists x.\, \xi'\{x = t\}\}}$$

However, using this rule instead of wit would make the system incomplete.

Theorem 9 (*MAUS vs. MAS*)**.** *For any* A *free of* $=$*-subformulas,* $1 \xrightarrow{\mathit{MAUS}} A$ *if and only if* $1 \xrightarrow{\mathit{MAS}} A$*.*

Sketch. One direction is simple, since the rules $\{\mathsf{refl}, \mathsf{cong}\}$ suffice to establish $t = t$, and the other rules of *MAS* are also derivable in *MAUS*. In the other direction, the *MAUS* rule ex is admissible in *MAS*. For applications of match in the *MAUS* proof, the relevant applications of wit must first be permuted below the rule, and the relevant applications of refl and cong (which do not interact with any other logical connective) must be permuted to above the instance of wit. These permutations are height-preserving, and the instances of cong and refl can only establish equalities. The entire argument then proceeds by induction on the height of the *MAUS* proof, splitting inductive cases from the bottom-most instance of match. □

MAUS has no equality rules that can introduce new variables in the premise of a rule. Thus, in the usual flex-flex case of unification, where Robinson's algorithm would allocate a fresh variable to instantiate both operands of the equality, in *MAUS* we would just instantiate one variable with the other. For the first-order case this is not problematic, but it does become an issue for higher-order terms. To illustrate, the higher-order pattern unification problem $\forall u, v, w.\, \exists X, Y.\, X\, u\, v = Y\, v\, w$ is solvable if we instantiate X with $\lambda x, y.\, Z\, y$ and Y with $\lambda x, y.\, Z\, x$ for some fresh existentially quantified variable Z, but there is no solution where one of X and Y is instantiated in terms of the other.

3.2 Disequations

Let us now turn to both positive and negative occurrences of equality, writing $(s \neq t)$ for $\overline{(s = t)}$. Before proceeding further, there is an important matter to settle: the linear nature of equations. When equality has only positive occurrences, it is immaterial whether we give it a linear semantics or not, but with negative occurrences this becomes an important choice. If we were to treat disequalities too as linear, then we must accept that formulas such as $(x = y) \multimap (x = y) \otimes (x = y)$ (which results from match on $(x = y) \multimap \mathsf{p}(x, x) \multimap \mathsf{p}(y, y)$) will be unprovable, since $(x \neq y)$ will be consumed by the first equation. Note that it is not sufficient to alter match to be additive, *i.e.*, to use & instead of $\otimes$ in the premise, as this still leaves the former formula unprovable.

If we had the full linear logic, we could use the exponential connectives to indicate that certain equalities are non-linear: $!(x = y) \multimap \mathsf{p}(x, x) \multimap \mathsf{p}(y, y)$. This does not work in *MALL*, so we instead add rules to treat disequalities as implicitly exponential with the following rules.

$$\dfrac{\xi\{\bot\}}{\xi\{s \neq t\}}\,\mathsf{qwk} \qquad \dfrac{\xi\{(s \neq t) \parr (s \neq t)\}}{\xi\{s \neq t\}}\,\mathsf{qct}$$

This is tantamount to assuming the derivability of $(s = t) \vdash 1$ and $(s = t) \vdash (s = t) \otimes (s = t)$, *i.e.*, of $(s = t) \dashv\vdash\, !(s = t)$ and $(s \neq t) \dashv\vdash\, ?(s \neq t)$. Note that this leaves the rest of the logic untouched; indeed, it is a conservative extension of *MAUS*.

An important property of the rules refl and cong of *MAUS* is that they simplify equations between two terms to a collection of equations on strict subterms. This suffices for the simple case of positive equations, but it is not complete in the general case. In particular, symmetry $(\forall x, y.\, (x = y) \multimap (y = x))$ and transitivity $(\forall x, y, z.\, (x = y) \multimap (y = z) \multimap (x = z))$, which are part of the axioms of equality, are not derivable in *MAUS*. Both these properties require rewriting an equation into other equations on the same or unrelated terms.

[2] We write $\mathrm{bvs}(\xi)$ for the variables bound in ξ that are visible at its hole.

Symmetry comes in two forms:

$$\frac{\xi\{t = s\}}{\xi\{s = t\}}\,\mathsf{sym} \qquad \frac{\xi\{t \neq s\}}{\xi\{s \neq t\}}\,\mathsf{nsym}$$

It turns out that these rules are stronger than strictly necessary, because the cong rule already reduces equations between arbitrary terms into equations where one operand is a variable. One can therefore limit sym to, for instance:

$$\frac{\xi\{x = t\}}{\xi\{t = x\}}$$

where t stands for any term and x stands for a variable. Indeed, in an implementation we can further orient sym by only rewriting if the term t is lpo-larger than x, where the variables are given some arbitrary ordering. We do not make these refinements in this paper to keep things simple.

Likewise, transitivity might appear to be a simple rule.

$$\frac{\xi\{(s = t) \otimes (t = u)\}}{\xi\{s = u\}} \tag{1}$$

However, this rule has the problem that it mentions a term t in the premise that is not necessarily in the conclusion. Inventing the interpolant t can also be arbitrarily expensive because the calculus does not constrain its shape in any way. Indeed, this is the only rule that breaks analyticity on terms. Fortunately, there is a simple fix: we observe that the rule corresponds to the entailment $(s = t) \otimes (t = u) \vdash (s = u)$, which is equivalent to its curried form $(t = u) \vdash (s = t) \multimap (s = u)$. Hence, we alter the rule to:

$$\frac{\xi\{t = u\}}{\xi\{(s \neq t) \parr (s = u)\}} \tag{2}$$

Now every term in the premise is also in the conclusion.

However, even this rule is not entirely incremental, because the term s has two occurrences in the conclusion, which requires a potentially expensive computation when applying the rule. We can modify it to just check that the two occurrences are equal:

$$\frac{\xi\{(s = u) \otimes (t = v)\}}{\xi\{(s \neq t) \parr (u = v)\}} \tag{3}$$

But, this form of rule, which is inter-derivable with (2), is just an instance of match for the $=$ predicate, and therefore redundant! In other words, transitivity of equality is just a special case of match.

This is not the full story: we also need to account for the transitivity of disequations, which is the contrapositive of the entailment $(s = t) \otimes (t = u) \vdash (s = u)$.

$$\frac{\xi\{(s \neq u)\}}{\xi\{(s \neq t) \parr (t \neq u)\}}$$

Once again, as t is repeated in the conclusion, we relax it to an equation to obtain the following rule, which is dual to (3).

$$\frac{\xi\{(t = u) \otimes (s \neq v)\}}{\xi\{(s \neq t) \parr (u \neq v)\}}\,\mathsf{trans}$$

Definition 10 (System *MAQS*)**.** Let *MAQS* stand for

$$\mathit{MAUS} \cup \{\mathsf{qwk}, \mathsf{qct}, \mathsf{sym}, \mathsf{nsym}, \mathsf{trans}\}$$

Theorem 11. *The transitivity rule (1) is admissible in* MAQS*.*

Sketch. By admissibility of cut and the *MAQS*-derivability of $\xi\{s = s\}$ for any ξ and s:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\xi\{(s = u) \otimes (u = t)\}}{\xi\{(u = s) \otimes (u = t)\}}\,\mathsf{sym}}{\xi\{(u \neq u) \parr (s = t)\}}\,\mathsf{match}}{\xi\{1 \otimes ((u \neq u) \parr (s = t))\}}}{\xi\{(u = u) \otimes ((u \neq u) \parr (s = t))\}}\,\text{derivability of } \xi\{u = u\}}{\xi\{s = t\}}\,\mathsf{cut},\,\mathsf{swr}$$

□

We will omit in this paper a formal soundness and completeness result for *MAQS* with respect to a sequent calculus. It is fairly easy to show by a straightforward induction that: (1) *MAQS* is a conservative extension of *MAUS*, and (2) that any *MAQS* derivation can be justified by a *MAUS* derivation where the end-formula is augmented by suitable instances of the following axioms.

Definition 12. The following (infinite set of) axioms characterize first-order equality.[3]

$$\forall x.\,(x = x), \qquad \forall x, y.\,(x = y) \multimap (y = x),$$
$$\forall x, y, z.\,(x = y) \otimes (y = z) \multimap (x = z),$$
$$\forall x, y.\,(x = y) \multimap \mathsf{1}, \qquad \forall x, y.\,(x = y) \multimap (x = y) \otimes (x = y),$$
$$\forall \vec{x}, \vec{y}.\,(\vec{x} = \vec{y}) \multimap \mathsf{f}(\vec{x}) = \mathsf{f}(\vec{y}),$$
$$\forall \vec{x}, \vec{y}.\,(\vec{x} = \vec{y}) \multimap \mathsf{p}(\vec{x}) \multimap \mathsf{p}(\vec{y}).$$

The last two axioms are schematic over f and p.

3.3 Intensional Equality

The system *MAQS* is too weak to show that $\mathsf{z} \neq \mathsf{s}(\mathsf{z})$ where z and s denote the natural number 0 and the successor operation. It also cannot reason fruitfully about disequations, such as to show that $(\mathsf{s}(x) = \mathsf{s}(\mathsf{s}(y))) \multimap (x = \mathsf{s}(y))$. This weakness is not problematic for mathematics, where equality is given an extensional or Leibniz interpretation, and any additional properties of function symbols are added as axioms. For example, both the above properties are instances of the Peano axioms for the natural numbers. However, in a computational setting where terms are used to encode data, we will generally treat equality as *intensional*, with functions interpreted as *term constructors*, *i.e.*, enjoying discrimination (applications of different function symbols are different) and injectivity (if two applications of the same function are equal, then the arguments must be equal). In the next section where we consider fixpoints, we will rely critically on this intensional view of equality.

Let us first consider the discrimination property. Once again, we have a choice of semantics. If we rewrite $\mathsf{z} \neq \mathsf{s}(\mathsf{z})$ to $\mathsf{1}$, then it does not let us prove, for instance, that $\mathsf{z} = \mathsf{s}(\mathsf{z}) \multimap \mathsf{0}$. In μ*MALL*, the corresponding sequent $\vdash \mathsf{z} \neq \mathsf{s}(\mathsf{z}), \mathsf{0}$ would be considered as provable because the set of unifiers of z and $\mathsf{s}(\mathsf{z})$ is empty. Therefore, we choose to use an additive semantics for discrimination, reflected in the following inference rule.

$$\frac{\xi\{\top\}}{\xi\{\mathsf{f}(\vec{s}) \neq \mathsf{g}(\vec{t})\}}\,\mathsf{disc}$$

Here, f and g denote different function symbols.

Related to discrimination is the topic of finiteness: because terms are assumed to be finite, a circular disequation should also be rewritten to $\top$. The prototypical scenario is the formula $(x = \mathsf{f}(x)) \multimap \mathsf{0}$, which in μ*MALL* is true because x and $\mathsf{f}(x)$ cannot be unified. To make this formal in the *COS*, we first define term contexts (τ):

$$\tau ::= \mathsf{f}(\vec{s}, \tau^*, \vec{t}) \qquad\qquad \tau^* ::= \Box \mid \tau$$

We will write $\tau\{t\}$ for the replacement of the single occurrence of the hole $\Box$ in τ with t. The finiteness rule is then;

$$\frac{\xi\{\top\}}{\xi\{x \neq \tau\{x\}\}}\,\mathsf{fin}$$

Note that this rule might appear to be non-incremental as it examines an entire term, but it can easily be made incremental by adding a *not-occurs-in* predicate on terms and giving it the obvious rules. We skip the straightforward elaboration of this basic idea in this paper.

Injectivity amounts to the reverse of the congruence axiom (Definition 12) for equality:

$$\forall \vec{x}, \vec{y}.\,\mathsf{f}(\vec{x}) = \mathsf{f}(\vec{y}) \multimap (\vec{x} = \vec{y}). \tag{4}$$

Its contrapositive is a natural *COS* rule for disequation:

$$\frac{\xi\{(\vec{s} \neq \vec{t})\}}{\xi\{\mathsf{f}(\vec{s}) \neq \mathsf{f}(\vec{t})\}}\,\mathsf{inj}$$

where $(\vec{s} \neq \vec{t})$ stands for $\overline{(\vec{s} = \vec{t})}$. Note that since we have qwk and qct, it is immaterial whether the premise uses $\parr$ or $\oplus$: the two choices are inter-derivable.

[3] These are the linear versions of the axioms from [13, p.236].

Definition 13 (System *MAIS*)**.** Let *MAIS* stand for

$$MAQS \cup \{\mathsf{disc}, \mathsf{fin}, \mathsf{inj}\}$$

It is interesting to note that we can separate finiteness from discrimination in *MAIS*. Dropping fin, for instance, allows *MAIS* to reason naturally about infinite terms.

Example 14. *Here is an example of discrimination interacting with transitivity in MAIS.*

$$\dfrac{\dfrac{\dfrac{1 \otimes \top}{(x = x) \otimes \top}\,\mathsf{refl}}{(x = x) \otimes (\mathsf{z} \neq \mathsf{s}(\mathsf{z}))}\,\mathsf{disc}}{(\mathsf{z} \neq x) \parr (x \neq \mathsf{s}(\mathsf{z}))}\,\mathsf{trans}$$

We can show that *MAIS* is sound and complete with respect to *QMALL*, which is the fragment of *μMALL* without fixpoints and higher-order terms. The rules of *QMALL* are given in Figure 4.

Theorem 15. *If* $A \xrightarrow{MAIS} B$*, then* $A \vdash B$ *in QMALL.*

Proof. Each inference rule of *MAIS* is derivable as an entailment in *QMALL*, similar to Theorem 3. □

In order to show completeness of *MAIS* with respect to *QMALL*, we will need to reason about the relation of term substitution and equations.

Definition 16. A *term substitution* $(\theta, \sigma, \dots)$ is a set of assignments of terms to variables, $[t_1/x_1, \dots, t_n/x_n]$, abbreviated as $\left[\vec{t}/\vec{x}\right]$. We write $\theta\mathcal{A}$ to stand for the simultaneous capture-avoiding replacement of the variables in the domain of θ by their assigned terms in the syntactic entity (term, formula, context, *etc*) $\mathcal{A}$. The *equational representation* of $\theta = \left[\vec{t}/\vec{x}\right]$, written Q_θ, is the formula $(\vec{x} = \vec{t})$.

Theorem 17. *The following rule is admissible in MAIS:*

$$\frac{\xi\{\theta A\}}{\xi\{\overline{Q_\theta} \parr A\}}$$

Sketch. By induction on the structure of A. Note that all (dis)equations in *MAIS* are reduced to (dis)equations involving variables, which are then combined with the disequations in $\overline{Q_\theta}$ using match or trans. □

Theorem 18. *If* A *is a true formula of QMALL, then* $1 \xrightarrow{MAIS} A$*.*

Proof. As before Theorem 4, we will show that the inference rules of *QMALL* are admissible in *MAIS*. There are only three interesting rules not accounted for already for *MALL*/*MAS*.

- The reflexivity rule of *QMALL*: $\dfrac{}{\vdash s = s}=$. This follows immediately by induction on the structure of s.
- The disequation rule of *QMALL*:

$$\frac{\text{for every } \theta \in \mathrm{csu}(s,t), \vdash \theta\Gamma}{\vdash \Gamma, s \neq t}$$

  If $\mathrm{csu}(s,t) = \emptyset$, then we are in exactly one of three possibilities: (1) s and t have different head function symbols, for which we use disc, (2) s and t have the same head function symbol but some of their arguments are point-wise non-unifiable, in which case we use inj and recurse, and (3) one of s or t is a variable and it occurs in the other, in which case we appeal to fin. If $\mathrm{csu}(s,t) \neq \emptyset$, then we can show by an induction on the structure of s and t that the following rule is admissible in *MAIS*:

$$\frac{\xi\{\overline{Q_{\mathrm{mgu}(s,t)}}\}}{\xi\{s \neq t\}}$$

  where mgu stands for the most general unifier for s and t, which exists (and is easy to compute) for first-order terms. We then appeal to Theorem 17. □

4 Fixpoints

In this section we will show how to build fixpoint reasoning on top of *MALL* intensional equality as defined in Section 3.3. We will follow the tradition of μ*MALL* [2] and *LINC* [19] and give least and greatest fixpoint semantics based on induction and coinduction rules. Case-branching and case-pruning in both these systems is performed by the rules for disequations. These proof systems are often also presented together with generic quantification (using the ∇-quantifier), which requires some additional reasoning about equality (known as *equivariance*) that is not part of *MAIS*; we leave this to future work.

To be concrete, we extend the grammar of predicates with two new De Morgan dual constructs: $\mu p(\vec{x}).\, A$ and $\nu p(\vec{x}).\, A$, where in each case the predicate symbol p and the variables $\vec{x} = x_1, \ldots, x_n$ are bound by the corresponding μ or ν in the formula body A. These predicates have the same arity as the number of elements of $\vec{x}$, and fixpoint atoms are intended to be equivalent to their *unfoldings*.

$$\begin{array}{lcl} (\mu p(\vec{x}).\, A)(\vec{s}) & \dashv\vdash & [\vec{s}/\vec{x}][\mu p(\vec{s}).\, A/p]A \\ (\mu p(\vec{x}).\, A)(\vec{s}) & \dashv\vdash & [\vec{s}/\vec{x}][\mu p(\vec{s}).\, A/p]A \end{array} \tag{5}$$

To simplify this paper, we will assume no mutual recursion between different fixpoint predicates; it is easy to extend the development to the general case.

For global consistency, which is equivalent to termination of cut-elimination, such fixpoint predicates must be *monotonic* [2, Definition 2.1]. (Strictly speaking, *MALL* does not require monotonicity as all fixpoint definitions have solutions, but monotonicity vastly simplifies the meta-theory of μ*MALL*.)

Definition 19. A fixpoint predicate (μ or ν) is monotonic if no bound predicate symbols are negated in it.

The De Morgan dual of a fixpoint atom is defined as follows:

$$\begin{aligned} \overline{(\mu p(\vec{x}).\, A)(\vec{s})} &= (\nu p(\vec{x}).\, \overline{[\overline{p}/p]A})(\vec{s}) \\ \overline{(\nu p(\vec{x}).\, A)(\vec{s})} &= (\mu p(\vec{x}).\, \overline{[\overline{p}/p]A})(\vec{s}) \end{aligned}$$

where the substitution $[\overline{p}/p]$ should be interpreted as replacements of atoms of the form $p(\vec{t})$ with $\overline{p(\vec{t})}$.

Example 20. *The following are standard inductive definitions of all (*nat*) and even (*evn*) natural numbers.*

$$\begin{aligned} \mathsf{nat} &\triangleq \mu \mathsf{nat}(n).\, (n = \mathsf{z}) \oplus (\exists m.\, (n = \mathsf{s}(m)) \otimes \mathsf{nat}(m)) \\ \mathsf{evn} &\triangleq \mu \mathsf{evn}(n).\, (n = \mathsf{z}) \oplus (\exists m.\, (n = \mathsf{s}(\mathsf{s}(m))) \otimes \mathsf{evn}(m)) \end{aligned}$$

Here, $\triangleq$ is just a notation for abbreviations that has no manifestation in the syntax of formulas or the proof system. We will illustrate the following pair of formulas as motivating examples (both are intended to be true).

1. $\forall n.\, \mathsf{nat}(\mathsf{s}(n)) \multimap \mathsf{nat}(n)$
2. $\forall n.\, \mathsf{nat}(n) \multimap (\mathsf{evn}(n) \oplus \mathsf{evn}(\mathsf{s}(n)))$

4.1 Finite Unfolding

Let us first consider of fixpoints, without giving any least or greatest fixpoint semantics to μ and ν. As (5) defines equivalences, they are readily transformed into *COS* rules:

$$\frac{\Xi\{[\vec{s}/\vec{x}][\mu p(\vec{x}).\, A/p]A\}}{\Xi\{(\mu p(\vec{x}).\, A)(\vec{s})\}}\,\mu\mathsf{unf} \qquad \frac{\Xi\{[\vec{s}/\vec{x}][\nu p(\vec{x}).\, A/p]A\}}{\Xi\{(\nu p(\vec{x}).\, A)(\vec{s})\}}\,\nu\mathsf{unf}$$

Example 21. *Theorem 1 from Example 20 is depicted in Figure 2. At the point marked $\star$, the problem has been reduced completely to a unification problem. Note that fixpoint predicates with arguments are treated as atomic formulas, so* match *is applicable.*

4.2 Induction and Coinduction

Finite unfolding is not strong enough to prove inductive theorems, such as case (2) from Example 20. In μ*MALL* [2] (which uses concepts from *LINC* [19]), such theorems are proved by the use of a specific coinduction rule:

$$\frac{\vdash \forall \vec{x}.\, (i(\vec{x}) \multimap [i/p]A) \quad \vdash \Gamma, i(\vec{s})}{\vdash \Gamma, (\nu p(\vec{x}).\, A)(\vec{s})}$$

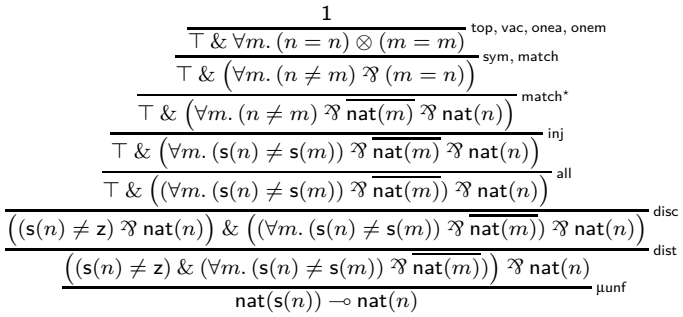

Figure 2: Example 21, case (1).

Here, i schematically represents an *invariant* of the right arity, which can itself be a fixpoint predicate. Note that the first premise is entirely isolated from the context Γ, and therefore functions as a side-condition to the rule.

This isolation from the context makes it difficult to import this rule into the *COS* tradition. The following is one possible formulation:

$$\frac{(\forall x.\, i(\vec{x}\,) \multimap [i/p]A) \otimes \xi\{i(\vec{s}\,)\}}{\xi\{(\nu p(\vec{x}\,).\, A)(\vec{s}\,)\}}$$

While sound, it is difficult to predict the impact such a non-contextual rule will have on the meta-theory. It is straightforward in the *COS* to give a meaning to a substitution of a *derivation* in a formula context by simply wrapping the context around each step of the derivation, but this kind of substitution would be disallowed by the above rule.

An alternative would be to isolate the extra premise using the exponentials of the full linear logic:

$$\frac{\xi\{!(\forall x.\, i(\vec{x}\,) \multimap [i/p]A) \otimes i(\vec{s}\,)\}}{\xi\{(\nu p(\vec{x}\,).\, A)(\vec{s}\,)\}}$$

This rule can be shown to be sound with respect to μLL [1, sec. 4.4.1], which is the extension of $\mu MALL$ with the exponential connectives. However, it is not satisfactory because it requires the exponential connectives that are largely redundant with fixpoint definitions in the first place. Moreover, the notion of *monotonicity* is not as straightforward for μLL as for $\mu MALL$.

The essential requirement of the coinduction rule is *isolation*, so the right question to ask is can we achieve isolation in a contextual form without exponentials? It is interesting to consider if this is possible in *MALL* itself, but we expect that it would require a non-trivial encoding of formulas. We will instead use a slight extension of *MALL* with a new *modal* connective $\dagger$ with a sole *COS* rule:

$$\frac{\xi\{1\}}{\xi\{\dagger\, 1\}}\,\mathsf{isol}$$

The syntax of formula contexts is extended with a new form, $\dagger\,\xi$, as well, so rules can be applied inside $\dagger$-formulas. However, because isol is the sole rule for $\dagger$, it is the case that no external formula can interact with the operand of $\dagger$.

The interpretation of this connective is the identity, *i.e.*, $\dagger\, A$ and A are intended to be equi-provable. However, the connective $\dagger$ also lacks a De Morgan dual, and hence it cannot be used as a cut-formula in the admissible general cut rule. The only purpose of this extremely limited connective is to implement a contextualized version of the coinduction rule:

$$\frac{\xi\{\dagger(\forall x.\, i(\vec{x}\,) \multimap [i/p]A) \otimes i(\vec{s}\,)\}}{\xi\{(\nu p(\vec{x}\,).\, A)(\vec{s}\,)\}}\,\mathsf{coind}$$

Definition 22 (System $\mu MAIS^*$)**.** Let $\mu MAIS^*$ stand for:

$$MAIS \cup \{\mu\mathsf{unf}, \mathsf{coind}, \mathsf{isol}\}$$

Fact 23. *$\mu MAIS^*$ is conservative over MAIS.* □

Theorem 24. *The* $\nu\mathsf{unf}$ *rule is derivable in $\mu MAIS^*$.*

Proof. Use $i(\vec{x}\,) \triangleq [(\nu p(\vec{x}\,).\, A)/p]A$ for coind. □

$$
(a)\quad \cfrac{\cfrac{\forall x.\, \Big(\dagger\big(\underbrace{\forall u.\, \overline{i(u)} \parr ((u \neq \mathsf{z}) \mathbin{\&} \forall v.\, (u \neq \mathsf{s}(v)) \parr i(v))}_{M}\big)\Big) \otimes \big(i(x) \parr (\mathsf{evn}(x) \oplus \mathsf{evn}(\mathsf{s}(x)))\big)}{\forall x.\, \Big(\dagger\big(\forall u.\, \overline{i(u)} \parr ((u \neq \mathsf{z}) \mathbin{\&} \forall v.\, (u \neq \mathsf{s}(v)) \parr i(v)\big) \otimes i(x)\Big) \parr (\mathsf{evn}(x) \oplus \mathsf{evn}(\mathsf{s}(x)))}\,\mathsf{swr}}{\forall x.\, \mathsf{nat}(x) \multimap (\mathsf{evn}(x) \oplus \mathsf{evn}(\mathsf{s}(x)))}\,\mathsf{coind}
$$

. . . .

$$
(b)\quad \cfrac{\cfrac{\cfrac{\cfrac{\forall u.\, (u = \mathsf{z}) \parr (u \neq \mathsf{z})}{\forall u.\, ((u = \mathsf{z}) \oplus \exists w.\, (u = \mathsf{s}(\mathsf{s}(w))) \otimes \mathsf{evn}(w)) \parr (u \neq \mathsf{z})}}{\forall u.\, \mathsf{evn}(u) \parr (u \neq \mathsf{z})}}{\forall u.\, (\mathsf{evn}(u) \oplus \mathsf{evn}(\mathsf{s}(u))) \parr (u \neq \mathsf{z})}}{\forall u.\, \overline{i(u)} \parr (u \neq \mathsf{z})}
$$

Figure 3: Example 21, case (2).

Example 25 (Partitioning the Naturals)**.** *Case (2) of Example. 20 can be proved in μMAIS* with the invariant* $i(x) \triangleq \overline{\mathsf{evn}(x)} \mathbin{\&} \overline{\mathsf{evn}(\mathsf{s}(x))}$*. The proof begins as in Figure 3, (a), and then the isolated subformula indicated by* M *is independently proved. It is a fairly straightforward matter to use* μunf *and the remaining rules of MAIS for it; one case is sketched in (b).*

Theorem 26. *A* $\dagger$*-free formula is provable in μMAIS* if and only if it is provable in μMALL.*

Sketch. Straightforward extension of the proofs of theorems 15 and 18. The new rules of *μMAIS** are all valid entailments or admissible rules in *μMALL*, assuming that $\dagger A$ is treated as equivalent to A, and the shallow coinduction, unfolding, and identity rules of *μMALL* can be simulated by *μMAIS**. □

4.3 Incremental Identity

The system *μMAIS** has one remaining defect in the match rule, which for fixpoint predicates has the following form:

$$
\cfrac{\xi\{(\vec{s} = \vec{t}\,)\}}{\xi\Big\{(\mu p(\vec{x}\,).\, A)(\vec{s}\,) \parr (\nu p(\vec{x}\,).\, \overline{[\overline{p}/p]A})(\vec{t}\,)\Big\}}
$$

The defect lies in the fact that A and $\overline{A}$ can be arbitrarily large formulas that must be found in different parts of the selected formula. This makes the rule inflexible and non-incremental, for it requires that the two fixpoint predicates be put in the right form earlier (lower) in the proof. In this section, we consider the question of whether this feature of match can be made incremental as well, in an analogous manner to the change from the aid rule of of *MAS* in comparison to match of *MAQS*.

In preparation for this modification, let us first consider the following variant of the init rule of *μMALL* [2, Figure 2.1]:

$$
\cfrac{\vdash \forall \vec{x}\,.\, ([\overline{i}\,/p]A \parr [i/p]B)}{\vdash (\mu p(\vec{x}\,).\, A)(\vec{s}\,), (\nu p(\vec{x}\,).\, B)(\vec{s}\,)}\,\mathsf{fix\text{-}init}
$$

where i is a fresh predicate symbol of arity $\|\vec{x}\,\|$ that is not free in the conclusion. It is easy to see that this rule is stronger than init by the admissibility of arbitrary identity in *μMALL*. Soundness is more involved. Consider the following derivation, where the invariant $i \triangleq \overline{\mu p(\vec{x}\,).\, A}$.

$$
\cfrac{\cfrac{\vdash [\overline{i}\,/p]A, [i/p]B}{\vdash \forall \vec{x}\,.\, i(\vec{x}\,) \multimap [i/p]B}\,\forall, \multimap, \mu \qquad \cfrac{}{\vdash (\mu p(\vec{x}\,).\, A)(\vec{s}\,), i(\vec{s}\,)}\,\mathsf{init}}{\vdash (\mu p(\vec{x}\,).\, A)(\vec{s}\,), (\nu p(\vec{x}\,).\, B)(\vec{s}\,)}\,\nu
$$

The unclosed left branch is proved for the specific i we chose. However, if we had a generic proof of the sequent with a free occurrence of p, then the particular instance with i substituted for p remains provable. Indeed, we can show by a simple induction that a *μMALL* proof remains a proof if an atomic formula in it substituted with an arbitrary formula.

Through the lens of the *COS*, the fix-init rule becomes:

$$
\cfrac{\xi\{\dagger(\forall \vec{x}.\, [\overline{q}\,/p]A \parr [q/p]B) \otimes (\vec{s} = \vec{t}\,)\}}{\xi\{(\mu p(\vec{x}\,).\, A)(\vec{s}\,) \parr (\nu p(\vec{x}\,).\, B)(\vec{t}\,)\}}\,\mathsf{fix\text{-}match}
$$

with the side condition that q is not captured by ξ. (This can be explicitly written with a second-order quantifier $\forall q.$ in an extension of the system of this paper to the second-order *MALL*.) Note that if p is not captured by ξ, then we can just reuse it and the substitution in one of the operands of the $\parr$ in the premise is trivial. The $\dagger$ pseudo-connective of the previous section is reused here to isolate the check for the compatibility of the two fixpont predicates. It prevents other information from ξ from interfering with its validity.

Definition 27. Let *μMAIS* stand for $\mu MAIS^* \cup \{\mathsf{fix\text{-}match}\}$, with the understanding that match is limited to atomic predicate symbols and bound predicate symbols alone.

Theorem 28. *The following rule is derivable in μMAIS:*

$$\frac{\xi\{(\vec{s} = \vec{t})\}}{\xi\{(\mu p(\vec{x}).\, A)(\vec{s}) \parr (\nu p(\vec{x}).\, \overline{A})(\vec{t})\}}$$

Sketch. Note that this is the analogue of the match rule for fixpoint predicates. The proof is immediate by $\mathsf{fix\text{-}match}$, isol, and the *μMAIS*-derivability of arbitrary identity:

$$\frac{\xi\{1\}}{\xi\{A \parr \overline{A}\}}.$$

□

Corollary 29. *The system μMAIS is sound and complete with respect to $\mu MAIS^*$.*

Proof. The argument for the soundness of $\mathsf{fix\text{-}match}$ in $\mu MAIS^*$ follows the outline of the similar argument for *μMALL* above. The cases for the remaining rules are trivial. □

5 Conclusion, Related Work, and Perspectives

We have now seen how to extend the calculus of structures for first-order multiplicative additive linear logic with a contextual and incremental treatment of intensional equality and inductive and coinductive fixpoints.

- *Background on μMALL:* We refer the interested reader to the comprehensive article [2] for the related literature on *μMALL*. Classical and intuitionistic versions of *μMALL* also exist [1], which can be adapted into the corresponding *COS* calculi using the techniques of this paper.
- *Equivariance:* The logic *μMALL* forms the basis of the Bedwyr model checker [3], and an intuitionistic variant (the logic $\mathcal{G}$) is the core of the Abella proof-assistant [10]. The main feature in these systems missing from *μMALL* is generic reasoning and nominal quantification (using the ∇-operator). Adding support for nominals to *MAIS* seems doable: we need to add rules for equivariance. These rules will be close to the actual implementation of equivariant unification that builds a *renumbering substitution* incrementally.
- *Second-order:* The *COS* has been extended to second-order propositional multiplicative linear logic [18]. There is a natural embedding of inductive and coinductive fixpoints into second-order *MALL* [2]. Second-order quantification also admits the powerful Leibniz equality. It would be interesting to see if we can give a contextual and incremental treatment to the second-order *MALL*. As mentioned in Section 4.3, using a second-order quantifier would simplify the $\mathsf{fix\text{-}match}$ rule, and make it even more incremental.
- *Meta-theory and focusing:* The system *μMAIS* is entirely compatible with the focused *COS* outlined in [8]. However, the meta-theory of the focused version of *μMAIS*, particularly an internal completeness proof, remains for future work. An internal cut-elimination proof for *μMAIS*, such as one using *splitting* [17], is also missing.
- *Other deep inference systems:* There are two main alternatives to the *COS* for deep inference: *nested sequents* [6, 11] and *open deduction* [12]. Unfortunately, even there the case of equality and fixpoints seems to be overlooked. At least for nested sequents it seems that the techniques outlined in sections 3 and 4 can be readily adapted. For open deduction, which further generalizes the *COS* to unite formulas and derivations in the same syntactic category, and admits connectives between derivations, the question of equality is obviously more complex.
- *Other treatments of equality:* For more mathematical uses of equality, the more standard treatment is based on *paramodulation* [15] rather than unification. While the paramodulation rule is certainly contextual, it is an interesting challenge to represent it in an incremental fashion. We expect the answer will be to construct a calculus of *equality propagation*, similar to the trans rule of Section 3.

- *Alternatives to coinduction:* Coinduction can also be done implicitly using cyclic proofs [5]. However, it is unclear how to represent cyclic proofs in their full generality in a unary system like the COS. The cyclic structure of a branching proof can be fairly complex and is easily obscured by the interleaving of the different branches in COS proofs.

  Instead of the fixpoint combinators μ and ν, we can also use definitions [16], which are generally more user-friendly than combinators and forms the basis of the two-level approach for computaional logic [10]. Definitions can be straightforwardly compiled to combinators, so the approach outlined in this paper should already be sufficient, but we could also address the definition-left rule more directly by compiling it to a sequence of $\neq$ rules.

- *Higher-order considerations:* In the higher-order case, where the relevant csus in the $\neq$ rule can have more than one element, the incremental approach of μMAIS can have a real payoff over the sequent calculus. Instead of forcefully duplicating the whole sequent for each member of the csu (which can be infinite in general), we would apply the steps of Huet's algorithm as rules to simplify the higher-order equality. In fact, μMAIS can serve as a language for proof certificates for higher-order unification. The steps of higher-order unification can be interleaved with ordinary logical reasoning. This is indeed already the approach taken by most systems that implement higher-order (pattern) unification: it is common to build a constraint management system for higher-order equality assumptions that cannot be immediately discharged [14].

Acknowledgement: Thanks to Dale Miller and Lutz Straßburger for many useful discussions. This work was partially funded by the ERC Advanced Grant *ProofCert*, and the *Demtech* grant from the Danish Council for Strategic Research.

References

[1] D. Baelde. *A linear approach to the proof-theory of least and greatest fixed points.* PhD thesis, Ecole Polytechnique, Dec. 2008.

[2] D. Baelde. Least and greatest fixed points in linear logic. *ACM Trans. on Computational Logic*, 13(1), Apr. 2012.

[3] D. Baelde, A. Gacek, D. Miller, G. Nadathur, and A. Tiu. The Bedwyr system for model checking over syntactic expressions. In F. Pfenning, editor, *21th Conf. on Automated Deduction (CADE)*, number 4603 in Lecture Notes in Artificial Intelligence, pages 391–397, New York, 2007. Springer.

[4] P. Brauner, C. Houtmann, and C. Kirchner. Principles of superdeduction. In *22th Symp. on Logic in Computer Science*, pages 41–50, 2007.

[5] J. Brotherston and A. Simpson. Sequent calculi for induction and infinite descent. *J. of Logic and Computation*, 21(6):1177–1216, Dec. 2011.

[6] K. Brünnler. *Nested Sequents.* Habilitationsschrift, Universität Bern, 2010.

[7] K. Chaudhuri. Subformula linking as an interaction method. In S. Blazy, C. Paulin-Mohring, and D. Pichardie, editors, *Proceedings of the 4th Conference on Interactive Theorem Proving (ITP)*, volume 7998 of *Lecture Notes in Computer Science*, pages 386–401. Springer, July 2013.

[8] K. Chaudhuri, N. Guenot, and L. Straßburger. The Focused Calculus of Structures. In *Computer Science Logic: 20th Annual Conference of the EACSL*, Leibniz International Proceedings in Informatics (LIPIcs), pages 159–173. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Sept. 2011.

[9] G. Dowek, T. Hardin, and C. Kirchner. Theorem proving modulo. *J. of Automated Reasoning*, 31(1):31–72, 2003.

[10] A. Gacek, D. Miller, and G. Nadathur. A two-level logic approach to reasoning about computations. *J. of Automated Reasoning*, 49(2):241–273, 2012.

[11] N. Guenot. *Nested Deduction in Logical Foundations for Computation.* Ph.d. thesis, Ecole Polytechnique, 2013.

[12] A. Guglielmi, T. Gundersen, and M. Parigot. A proof calculus which reduces syntactic bureaucracy. In C. Lynch, editor, *Proceedings of the 21st International Conference on Rewriting Techniques and Applications*, volume 6 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 135–150, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[13] J. Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, 1st edition, 2009.

[14] J. Reed. Higher-order constraint simplification in dependent type theory. In J. Cheney and A. P. Felty, editors, *Proceedings of the Fourth International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice (LFMTP)*, pages 49–56. ACM, 2009.

[15] G. Robinson and L. Wos. Paramodulation and theorem-proving in first-order theories with equality. *Machine Intelligence*, 4:135–150, 1969.

[16] P. Schroeder-Heister. Rules of definitional reflection. In M. Vardi, editor, *8th Symp. on Logic in Computer Science*, pages 222–232. IEEE Computer Society Press, IEEE, June 1993.

[17] L. Straßburger. *Linear Logic and Noncommutativity in the Calculus of Structures*. PhD thesis, Technische Universität Dresden, 2003.

[18] L. Straßburger. Some observations on the proof theory of second order propositional multiplicative linear logic. In P.-L. Curien, editor, *Typed Lambda Calculi and Applications, TLCA'09*, volume 5608 of *Lecture Notes in Computer Science*, pages 309–324. Springer, 2009.

[19] A. Tiu. *A Logical Framework for Reasoning about Logical Specifications*. PhD thesis, Pennsylvania State University, May 2004.

$\underline{MALL}$

$$\frac{}{\vdash a, \overline{a}}\,\mathsf{init} \qquad \left[\frac{\vdash \Gamma, a \quad \vdash \Delta, \overline{a}}{\vdash \Gamma, \Delta}\,\mathsf{cut}\right] \qquad \frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\otimes \qquad \frac{}{\vdash 1}\,1 \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \bot}\,\bot \qquad \frac{\vdash \Gamma, A_i}{\vdash \Gamma, A_1 \oplus A_2}\,\oplus \qquad \frac{\vdash \Gamma, [t/x]A}{\vdash \Gamma, \exists x.\, A}\,\exists$$

$$\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\,\parr \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \bot}\,\bot \qquad \frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, A \,\&\, B}\,\& \qquad \frac{}{\vdash \Gamma, \top}\,\top \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, \forall x.\, A}\,\forall$$

. . . . . .

$\underline{QMALL} = MALL \cup$

$$\frac{}{\vdash t = t}= \qquad \frac{\text{for every } \theta \in \mathrm{csu}(s,t), \vdash \theta\Gamma}{\vdash \Gamma, s \neq t}\neq$$

. . . . . .

$\underline{\mu MALL} = QMALL \cup$

$$\frac{\vdash \Gamma, [\vec{s}/\vec{x}, (\mu p(\vec{x}).\, A)/p]A}{\vdash \Gamma, (\mu p(\vec{x}).\, A)(\vec{s})}\,\mu \qquad \frac{\vdash \Gamma, i(\vec{s}) \quad \vdash \forall x.\, (i(\vec{x}) \multimap [i/p]A)}{\vdash \Gamma, (\nu p(\vec{x}).\, A)(\vec{s})}\,\nu$$

Notes: in the $\oplus$ rule, $i \in \{1, 2\}$; in the $\forall$ rule, x is not free in Γ; the cut rule is admissible, but not part of the system; and this proof system is primarily drawn from [2, Figure 1].

Figure 4: Sequent calculus for $\mu MALL$

$\underline{MAX}$

$$\frac{\xi\{(A \parr B) \parr C\}}{\xi\{A \parr (B \parr C)\}}\,\mathsf{assoc} \qquad \frac{\xi\{B \parr A\}}{\xi\{A \parr B\}}\,\mathsf{comm} \qquad \frac{\xi\{(A \parr C) \otimes B\}}{\xi\{(A \otimes B) \parr C\}}\,\mathsf{swl} \qquad \frac{\xi\{B \otimes (A \parr C)\}}{\xi\{A \parr (B \otimes C)\}}\,\mathsf{swr} \qquad \frac{\xi\{1\}}{\xi\{1 \otimes 1\}}\,\mathsf{onem}$$

$$\frac{\xi\{A\}}{\xi\{A \oplus B\}}\,\mathsf{chl} \qquad \frac{\xi\{B\}}{\xi\{A \oplus B\}}\,\mathsf{chr} \qquad \frac{\xi\{[t/x]A\}}{\xi\{\exists x.\, A\}}\,\mathsf{wit} \qquad \frac{\xi\{(A \parr B) \,\&\, (A \parr C)\}}{\xi\{A \parr (B \,\&\, C)\}}\,\mathsf{dist} \qquad \frac{\xi\{\top\}}{\xi\{A \parr \top\}}\,\mathsf{abs} \qquad \frac{\xi\{1\}}{\xi\{\top\}}\,\mathsf{top} \qquad \frac{\xi\{1\}}{\xi\{1 \,\&\, 1\}}\,\mathsf{onea}$$

$$\frac{\xi\{A\}}{\xi\{A \parr \bot\}}\,\mathsf{bot} \qquad \frac{\xi\{\forall x.\, (A \parr B)\}}{\xi\{A \parr \forall x.\, B\}}\,\mathsf{all} \qquad \frac{\xi\{1\}}{\xi\{\forall x.\, 1\}}\,\mathsf{vac} \qquad \vdots \qquad \underline{MAS} = MAX \cup \left\{\frac{\xi\{1\}}{\xi\{a \parr \overline{a}\}}\,\mathsf{aid}\right\}$$

. . . . . .

$\underline{MAUS} = MAX \cup$

$$\frac{\xi\{(\vec{s} = \vec{t})\}}{\xi\left\{\mathsf{p}(\vec{s}) \parr \overline{\mathsf{p}(\vec{t})}\right\}}\,\mathsf{match} \qquad \frac{\xi\{1\}}{\xi\{x = x\}}\,\mathsf{refl} \qquad \frac{\xi\{(\vec{s} = \vec{t})\}}{\xi\{\mathsf{f}(\vec{s}) = \mathsf{f}(\vec{t})\}}\,\mathsf{cong} \qquad \frac{\xi\{\exists x.\, (A \parr B)\}}{\xi\{A \parr \exists x.\, B\}}\,\mathsf{ex}$$

. . . . . .

$\underline{MAQS} = MAUS \cup$

$$\frac{\xi\{\bot\}}{\xi\{s \neq t\}}\,\mathsf{qwk} \qquad \frac{\xi\{(s \neq t) \parr (s \neq t)\}}{\xi\{s \neq t\}}\,\mathsf{qct} \qquad \frac{\xi\{t = s\}}{\xi\{s = t\}}\,\mathsf{sym} \qquad \frac{\xi\{t \neq s\}}{\xi\{s \neq t\}}\,\mathsf{nsym} \qquad \frac{\xi\{(t = u) \otimes (s \neq v)\}}{\xi\{(s \neq t) \parr (u \neq v)\}}\,\mathsf{trans}$$

. . . . . .

$\underline{MAIS} = MAQS \cup$

$$\frac{\xi\{\top\}}{\xi\{\mathsf{f}(\vec{s}) \neq \mathsf{g}(\vec{t})\}}\,\mathsf{disc} \qquad \frac{\xi\{\top\}}{\xi\{x \neq \tau\{x\}\}}\,\mathsf{fin} \qquad \frac{\xi\{(\vec{s} \neq \vec{t})\}}{\xi\{\mathsf{f}(\vec{s}) \neq \mathsf{f}(\vec{t})\}}\,\mathsf{inj}$$

. . . . . .

$\underline{\mu MAIS^*} = MAIS \cup$

$$\frac{\xi\{[\vec{s}/\vec{x}, (\mu p(\vec{x}).\, A)/p]A\}}{\xi\{(\mu p(\vec{x}).\, A)(\vec{s})\}}\,\mu\mathsf{unf} \qquad \frac{\xi\{\dagger(\forall x.\, i(\vec{x}) \multimap [i/p]A) \otimes i(\vec{s})\}}{\xi\{(\nu p(\vec{x}).\, A)(\vec{s})\}}\,\mathsf{coind} \qquad \frac{\xi\{1\}}{\xi\{\dagger\, 1\}}\,\mathsf{isol}$$

. . . . . .

$\mu MAIS = \mu MAIS^* \cup$

$$\frac{\xi\{\dagger(\forall \vec{x}.\, [\overline{q}/p]A \parr [q/p]B) \otimes (\vec{s} = \vec{t})\}}{\xi\{(\mu p(\vec{x}).\, A)(\vec{s}) \parr (\nu p(\vec{x}).\, B)(\vec{t})\}}\,\mathsf{fix\text{-}match}$$

Notes: in wit, the substitution $[t/x]A$ is capture-avoiding; in all and ex, x is not free in A; in $\mu MAIS^*$, the match rule is understood to apply to (free or bound) predicate symbols and fixpoint predicates; and In $\mu MAIS$, the match rule is understood to apply to (free or bound) predicate symbols only, and q is a predicate symbol that is not captured by ξ.

Figure 5: The *COS* systems comprising $\mu MAIS^*$ and $\mu MAIS$