Peano Arithmetic and $\bar{\bar{\mu}}$MALL

Matteo Manighetti
Inria Saclay

Dale Miller
Inria Saclay and LIX, Institut Polytechnique de Paris

1 Introduction

We propose to examine some of the proof theory of arithmetic by using three proof systems. A linearized version of arithmetic, named $\bar{\bar{\mu}}$MALL, is MALL plus logical connectives to treat first-order term structures: equality and inequality, first-order universal and existential quantification, and the least and greatest fixed point operators.[1] The proof system $\bar{\bar{\mu}}$LKp is an extension of $\bar{\bar{\mu}}$MALL in which contraction and weakening are permitted and $\bar{\bar{\mu}}$LKp$^+$ is a further extension in which the cut rule is permitted. As their names implie, $\bar{\bar{\mu}}$LKp and $\bar{\bar{\mu}}$LKp$^+$ involve polarized classical formulas, as defined below.

It is known that $\bar{\bar{\mu}}$MALL has a cut-elimination result and is therefore consistent [3, 2]. We will show that $\bar{\bar{\mu}}$LKp is consistent by embedding it into second-order linear logic. We also show that $\bar{\bar{\mu}}$LKp$^+$ contains Peano arithmetic and that in a couple of different situations, $\bar{\bar{\mu}}$LKp is conservative over $\bar{\bar{\mu}}$MALL. Finally, we show that a proof that a relation represents a total function can be turned into a proof-search-based algorithm to compute that function.

Since we are interested in using $\bar{\bar{\mu}}$MALL to study *arithmetic*, we use first-order structures to encode natural numbers and fixed points to encode relations among numbers. This focus is in contrast to uses of the propositional subset of $\bar{\bar{\mu}}$MALL as a typing systems (see, for example, [7]). We shall limit ourselves to using invariants to reason about fixed points instead of employing other methods, such as infinitary proof systems (e.g., [4]) and cyclic proof systems (e.g., [17, 6]).

1.1 Polarized and unpolarized formulas

Following Church's Simple Theory of Types [5], we shall view the formulas and terms of arithmetic as simply typed λ-terms using the primitive types o and i, respectively. Propositional connectives have the usual typing: $\mathrm{o} \to \mathrm{o} \to \mathrm{o}$ for binary connectives and o for the units. There are six connectives that have types involving i, namely, $=$ and $\neq$, both of type $i \to i \to \mathrm{o}$; $\forall$ and $\exists$, both of type $(i \to \mathrm{o}) \to \mathrm{o}$; and μ and ν, both of type $((i \to \mathrm{o}) \to (i \to \mathrm{o})) \to (i \to \mathrm{o})$. These latter two connectives denote the least and greatest fixed point operators for one argument: additional such operators can easily be added to handle arities more than 1.

Formulas in our development of arithmetic are divided into two classes. Neither class will have atomic formulas, i.e., there are no (undefined) predicates. *Unpolarized* formulas are built using $\wedge$, tt, $\vee$, ff, $=$, $\neq$, $\forall$, $\exists$, μ, and ν. *Polarized* formulas are built using $\otimes$, 1, $\parr$, $\perp$, &, $\top$, $\oplus$, 0, $=$, $\neq$, $\forall$, $\exists$, μ, and ν. Note that the six connectives that have i in their typing are used in both of these classes: these are also the connectives that have an unambiguous polarity (we discuss polarity below).

As defined, both polarized and unpolarized formulas are in *negation normal form* in the sense that they contain no occurrences of negation. For convenience, we will occasionally allow implications in unpolarized formulas: in those cases, we treat $P \supset Q$ as $\overline{P} \vee Q$ where $\overline{P}$ is the result of replacing every

[1] The $\bar{\bar{\mu}}$MALL systems here is the system μMALL$^=$ in [3]: we have only changed the typesetting of its name.

Submitted to:
Linearity & TLLA 2022

occurrence of the logical connectives in P with its De Morgan dual (following the usual conventions for classical and linear logics and where $=$ and $\neq$ are duals, as are μ and ν).

1.2 Polarity of formulas

The connectives used in polarized formulas are given a *polarity*. The connectives ⅋, $\bot$, &, $\top$, $\forall$, $\neq$, and ν are *negative* while their De Morgan duals are *positive*. A polarized formula is positive or negative depending only on the polarity of its top-most connective.

A polarized formula $\hat{Q}$ is a *polarized version* of the unpolarized formula Q if every occurrence of & and $\otimes$ in $\hat{Q}$ is replaced by $\wedge$ in Q, every occurrence of ⅋ and $\oplus$ in $\hat{Q}$ is replaced by $\vee$ in Q, every occurrence of 1 and $\top$ in $\hat{Q}$ is replaced by tt in Q, and every occurrence of 0 and $\bot$ in $\hat{Q}$ is replaced by $f\!f$ in Q. Notice that if Q has n occurrences of propositional connectives, then there are 2^n formulas $\hat{Q}$ that are polarized versions of Q.

Fixed point expression, such as $((\mu\lambda P\lambda x(B\ P\ x))\ t)$, introduce variables of predicate type (here, P). In the case of the μ fixed point, any expression built using that predicate variable will be considered to be polarized positively. If the ν operator is used instead, any expressions built using the predicate variables it introduces is considered to be polarized negatively.

A formula is *purely positive* (resp., *purely negative*) if every logical connective it contains is positive (resp., negative). We generalize the familiar arithmetical hierarchy notation by using it to classify polarized formulas as follows. The Σ_1-formulas are exactly the purely positive formulas, and the Π_1-formulas are exactly the purely negative formulas. More generally, for $n \geq 1$, the Π_{n+1}-formulas are those negative formulas for which every positive subformula occurrence is a Σ_n-formula. Similarly, the Σ_{n+1}-formulas are those positive formulas for which every negative subformula occurrence is a Π_n-formula. A formula in Σ_n or in Π_n has at most $n-1$ alternations of polarity. Clearly, the dual of a Σ_n-formula is a Π_n-formula, and vice versa. We shall also extend these classifications of formulas to abstractions over terms: thus, we say that the term $\lambda x.B$ of type $i \to o$ is in Σ_n if B is a Σ_n-formula.

2 Linear and classical proof systems for polarized formulas

The $\bar{\mu}$MALL proof system [3, 2] for polarized formulas is the one-sided sequent calculus proof system given in Figure 1. The variable y in the $\forall$-introduction rule is an *eigenvariable*: it is restricted to not be free in any formula in the conclusion of that rule. The application of a substitution θ to a signature Σ (written $\Sigma\theta$ in the $\neq$ rule in Figure 1) is the signature that results from removing from Σ the variables in the domain of θ and adding back any variable that is free in the range of θ. In the $\neq$-introduction rule, if the terms t and t' are not unifiable, the premise is empty and immediately proves the conclusion.

The choice of using Church's λ-notation provides an elegant treatment of higher-order substitutions (needed for handing induction invariants) and provides a simple treatment of fixed point expressions and the binding mechanisms used there. In particular, we shall assume that formulas in sequents are always treated modulo $\alpha\beta\eta$-conversion. We usually display formulas in $\beta\eta$-long normal form when presenting sequents. Note that formula expressions such as $B\ S\ \vec{t}$ (see Figure 1) are parsed as $(\cdots((B\ S)t_1)\cdots t_n)$ if $\vec{t}$ is the list of terms $t_1,\ldots,t_n$.

If we were working in a two-sided calculus, the ν-rule in Figure 1 could be written in the two ways:

$$\frac{\Gamma \vdash \Delta, S\vec{t} \quad S\vec{x} \vdash BS\vec{x}}{\Gamma \vdash \nu B\vec{t}, \Delta}\ \textit{coinduction} \qquad \frac{\Gamma, S\vec{t} \vdash \Delta \quad BS\vec{x} \vdash S\vec{x}}{\Gamma, \mu B\vec{t} \vdash \Delta}\ \textit{induction}$$

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

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

$$\frac{\{\ \vdash \Gamma\theta : \theta = mgu(t,t')\ \}}{\vdash \Gamma, t \neq t'}\ \neq \qquad \frac{}{\vdash t = t}\ = \qquad \frac{\vdash \Gamma, Pt}{\vdash \Gamma, \exists x.Px}\ \exists \qquad \frac{\vdash \Gamma, Py}{\vdash \Gamma, \forall x.Px}\ \forall$$

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

Figure 1: The inference rules for the $\bar{\bar{\mu}}$MALL proof system

$$\frac{\vdash \Gamma, B(\nu B)\vec{t}}{\vdash \Gamma, \nu B\vec{t}}\ \mathit{unfold} \qquad \frac{\vdash \Gamma, Q, Q}{\vdash \Gamma, Q}\ C \qquad \frac{\vdash \Gamma}{\vdash \Gamma, Q}\ W \qquad \frac{\vdash \Gamma, Q \quad \vdash \Delta, \overline{Q}}{\vdash \Gamma, \Delta}\ \mathit{cut}$$

Figure 2: Some additional rules

That is, the one rule for ν yields both coinduction and induction. In general, we shall speak of the higher-order substitution term S used in both of these rules as the *invariant* of that rule (*i.e.*, we will not use the term co-invariant even though that might be more appropriate in some settings).

We make the following observations about this proof system.

1. The $\mu\nu$ rule is a limited form of the initial rule. The general form of the initial rule, namely, that the sequent $\vdash Q, \overline{Q}$ is provable, is admissible.
2. The μ rule allows the μ fixed point to be unfolded. This rule captures, in part, the identification of μB with $B(\mu B)$; that is, μB is a fixed point of B. This inference rule allows one occurrence of B in (μB) to be expanded to two occurrences of B in $B(\mu B)$. In this way, unbounded behavior can appear in $\bar{\bar{\mu}}$MALL where it does not occur in MALL.
3. The *unfold* rule in Figure 2, which simply unfolds ν-expression, is admissible in $\bar{\bar{\mu}}$MALL by using the ν-rule with the invariant $S = B(\nu B)$.
4. The weakening and contraction rules are admissible in $\bar{\bar{\mu}}$MALL for purely negative formulas.
5. The proof rules for equality guarantee that function symbols are all treated injectively: thus, function symbols will act only as term constructors.

We define $\bar{\bar{\mu}}$LKp by adding to be the proof system $\bar{\bar{\mu}}$MALL the rules for contraction C and weakening W (Figure 2). In addition, we define $\bar{\bar{\mu}}$LKp$^+$ to be $\bar{\bar{\mu}}$LKp with the cut rule added (also in Figure 2).

Example 1. *The formula* $\forall x \forall y[x = y \vee x \neq y]$ *can be polarized as either*

$$\forall x \forall y[x = y \parr x \neq y] \quad \mathit{or} \quad \forall x \forall y[x = y \oplus x \neq y].$$

Only the first of these is provable in $\bar{\bar{\mu}}$MALL, although both formulas are provable in $\bar{\bar{\mu}}$LKp.

3 $\bar{\bar{\mu}}$LKp$^+$ and Peano Arithmetic

While the cut rule (Figure 2) is admissible in $\bar{\bar{\mu}}$MALL [2], it is currently open as to whether or not the cut rule is admissible in $\bar{\bar{\mu}}$LKp. We conjecture that $\bar{\bar{\mu}}$LKp and $\bar{\bar{\mu}}$LKp$^+$ prove the same sequents. We can prove, however, that $\bar{\bar{\mu}}$LKp is consistent.

Theorem 1. *$\bar{\bar{\mu}}$LKp is consistent.*

The proof of this theorem can be found in Appendix A. It is worth noting that adding contraction to some logical systems with weak forms of fixed points can change that logic from being consistent to inconsistent. For example, both Girard [11] and Schroeder-Heister [16] describe a variant of linear logic with unfolding fixed points that is consistent, but when contraction is added, it becomes inconsistent. In their case, negations are allowed in the body of fixed point definitions. The theorem above proves that adding contraction to $\bar{\bar{\mu}}$MALL does not lead to inconsistency.

In order to show that Peano arithmetic is contained in $\bar{\bar{\mu}}$LKp$^+$, we need to deal with the following three aspects of logic.

Terms We introduce the primitive type i and the term-level signature $\{z\colon i, s\colon i \to i\}$, for zero and successor. We shall write numerals in bold, that is, $\mathbf{0}$, $\mathbf{1}$, $\mathbf{2}$, etc are abbreviations for z, $(s\ z)$, $(s\ (s\ z))$, etc. We also introduce an abbreviation for the predicate that holds only for such numerals.

$$nat = \mu\lambda N\lambda n(n = \mathbf{0} \oplus \exists m(n = (s\ m) \otimes N\ m))$$

Formulas We define the mapping $(\cdot)^\circ$ that translates formulas in Peano arithmetic into polarized formulas. The propositional connectives $\wedge$, tt, $\vee$, ff are mapped to polarized versions, say, $\oplus$, 1, $\oplus$, 0, respectively. The connectives $=$ and $\neq$ map to themselves. The first-order quantifiers are mapped so that they become explicitly typed, as follows. Recall that in Church's STT representation of quantified formulas, the universally quantified formula $\forall x.B$ is an abbreviation for $\forall(\lambda x.B)$: here, $\forall$ is a constant of type $(i \to o) \to o$. Similarly, the existential quantifier is coded by the constant $\exists$ of the same type. The function $(\cdot)^\circ$ replaces every occurrence of $\forall$ with $\lambda B.\forall x\ (\overline{nat\, x} \parr (Bx))$ and every occurrence of $\exists$ with $\lambda B.\exists x\ (nat\, x \otimes (Bx))$.

Proofs Peano Arithmetic is usually presented as a theory consisting of the following axioms and axiom scheme.

$$\begin{array}{ll}
\forall x.\ (sx) \neq z & \forall x.\ \forall y(x + sx) = s(x+y) \\
\forall x \forall y.\ (sx = sy) \supset (x = y) & \forall x.\ (x \cdot z = z) \\
\forall x.\ (x + z = x) & \forall x.\ (x \cdot sy = (x \cdot y + x)) \\
(Az \wedge \forall x.\ (Ax \supset A(s\ x))) \supset \forall x.Ax &
\end{array}$$

Since we wish to avoid introducing the extra constructors $+$ and $\cdot$, we encode addition and multiplications as relations. We can then extend the translation $(\cdot)^\circ$ to include

$$(x + y = w)^\circ := plus\, x\, y\, w \qquad \text{and} \qquad (x \cdot y = w)^\circ := mult\, x\, y\, w,$$

using the following fixed point definitions.

$$plus = \mu\lambda P\lambda n\lambda m\lambda p((n = z \otimes m = p) \oplus \exists n' \exists p'(n = (s\ n') \otimes p = (s\ p') \otimes P\ n'\ m\ p'))$$

$$mult = \mu\lambda M\lambda n\lambda m\lambda p\big((n = z \otimes p = z) \oplus \exists n' \exists p'(n = (s\ n') \otimes plus\ m\ p'\ p \otimes M\ n'\ m\ p')\big)$$

Our reusing of the familiar notation for the arithmetic hierarchy for classifying polarized formula is partially justified in the following sense: for all $n \geq 1$, if B is an unpolarized Π_n-formula then B° is Π_n, and if B is an unpolarized Σ_n-formula then B° is Σ_n.

Theorem 2 ($\bar{\bar{\mu}}$LKp$^+$ contains Peano arithmetic)**.** *Let Q be any unpolarized formula and let $\hat{Q}$ be a polarized version of Q. If Q is provable in Peano arithmetic then $(\hat{Q})^\circ$ is provable in $\bar{\bar{\mu}}$LKp$^+$.*

Proof. It is easy to prove that *mult* and *plus* describe precisely the multiplication and addition operations on natural numbers. Furthermore, the translations of the Peano Axioms can all be proved in $\bar{\mu}$LKp. We illustrate just one of these axioms here: a polarization of the translation of the induction scheme is

$$\overline{\big(Az \otimes \forall x.\ (\overline{nat\ x} \parr \overline{Ax} \parr A(s\ x))\big)} \parr \forall x.\ (\overline{nat\ x} \parr Ax)$$

An application of the ν rule to the second occurrence of $\overline{nat\ x}$ can provide an immediate proof of this axiom. Finally, the cut rule in $\bar{\mu}$LKp$^+$ allows us to encode the inference rule of modus ponens. □

4 Conservativity results for linearized arithmetic

The following theorem is our first conservativity result.

Theorem 3. *$\bar{\mu}$LKp is conservative over $\bar{\mu}$MALL for Σ_1-formulas and Π_1-formulas. In particular, let B be a either a Σ_1 or a Π_1-formula. Then $\vdash B$ has a $\bar{\mu}$LKp proof if and only if $\vdash B$ has a $\bar{\mu}$MALL proof.*

The case for Σ_1-formulas is proved by a straightforward argument about the permutation of proof rules for $\bar{\mu}$LKp. The case for Π_1-formulas has a simpler proof since weakening and contraction are admissible rules in $\bar{\mu}$MALL for Π_1-formulas.

Note that it is clear that if there exists a $\bar{\mu}$MALL proof of a purely positive formula, then that proof does not contain the ν rule, i.e., it does not contain the induction rule. Finally, given that first-order Horn clauses can interpret Turing machines [18], and given that Horn clauses can easily be encoded using purely positive formulas, it is undecidable whether or not a purely positive expression has a $\bar{\mu}$MALL proof. Similarly, purely positive formulas can be used to specify any general recursive function.

Our next conservativity result requires restricting the complexity of invariants used in the induction rule ν. We say that a sequent has a $\bar{\mu}$LKp(Σ_1) proof if it has a $\bar{\mu}$LKp proof in which all invariants of the proof are purely positive. This fragment is similar to the fragment $I\Sigma_1$ of Peano Arithmetic. A well-known result in the study of arithmetic is the following.

> Peano arithmetic is Π_2-conservative over Heyting arithmetic: if Peano arithmetic proves a Π_2-formula A, then A is already provable in Heyting arithmetic [8].

This result inspires the following theorem.

Theorem 4. *$\bar{\mu}$LKp(Σ_1) is conservative over $\bar{\mu}$MALL for Π_2-formulas. That is, if B is a Π_2-formula such that $\vdash B$ has a $\bar{\mu}$LKp(Σ_1) proof, then $\vdash B$ has a $\bar{\mu}$MALL proof.*

A proof for this theorem can be found in Appendix B.

Example 2. *Example 1 lists two polarized formulas. The formula $\forall x \forall y[x = y \parr x \neq y]$ is Π_2 and is provable in both $\bar{\mu}$MALL and $\bar{\mu}$LKp, while the formula $\forall x \forall y[x = y \oplus x \neq y]$ is Π_3 and is provable in $\bar{\mu}$LKp but not in $\bar{\mu}$MALL.*

5 Using proof search to compute functions

One way to prove that a binary relation ϕ encodes a function is to prove the *totality* and *determinancy* properties of ϕ: that is, prove $[\forall x \exists y.\phi(x,y)] \wedge [\forall x \forall y_1 \forall y_2.\phi(x,y_1) \supset \phi(x,y_2) \supset y_1 = y_2]$.

Clearly, these properties imply that for every natural number x, the predicate $\lambda y.\phi(x,y)$ denotes a singleton set. If our logic contains a choice operator, such as Church's *definite description* operator ι [5], then this function can be represented via the expression $\lambda x.\iota y.\phi(x,y)$. A more computationally-oriented approach to encoding such functions follows the Curry-Howard approach of relating proof theory to computation [12]: one extracts from a natural deduction proof of $\forall x \exists y.\phi(x,y)$ a λ-term, which can be seen as an algorithm for computing the implied function. The algorithmic content of such a λ-term arises from a non-deterministic rewriting process that iteratively selects β-redexes for reduction. In most typed λ-calculus systems, all such sequences of rewritings will end in the same normal form, although some sequences of rewrites might be very long, and others can be very short. This section will describe an alternative mechanism for computing functions from their relational specification that relies on using proof search mechanisms instead of the Curry-Howard correspondence.

Note that if P and Q are predicates of arity one and if P denotes a singleton, then $\exists x[Px \wedge Qx]$ and $\forall x[Px \supset Qx]$ are logically equivalent. We assume here that Px is a purely positive expression with x as its only free variable. Notice that the proof search semantics of these equivalent formulas are surprisingly different. In particular, if we attempt to prove $\exists x[Px \wedge Qx]$, then we must *guess* a term t and then *check* that t denotes the element of the singleton (by proving $P(t)$). In contrast, if we attempt to prove $\forall x[Px \supset Qx]$ then we allocate an eigenvariable y (which we will eventually instantiate with t) and then attempt to prove the sequent $\vdash Py \supset Qy$. Such an attempt at building a proof might actually *compute* the value t (especially if we can restrict proofs of that implication to not involve the general form of induction).

Example 3. *The following derivation verifying that 4 is a sum of 2 and 2.*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \mathbf{2} = (s\ \mathbf{1})}{=} \qquad \dfrac{}{\vdash \mathbf{4} = (s\ \mathbf{3})}{=} \qquad \vdash plus\ \mathbf{1}\ \mathbf{2}\ \mathbf{3}}{\vdash \mathbf{2} = (s\ \mathbf{1}) \otimes \mathbf{4} = (s\ \mathbf{3}) \otimes plus\ \mathbf{1}\ \mathbf{2}\ \mathbf{3}}\ \otimes \times 2}{\vdash \exists n' \exists p' (\mathbf{2} = (s\ n') \otimes \mathbf{4} = (s\ p') \otimes plus\ n'\ \mathbf{2}\ p')}\ \exists \times 2}{\vdash (\mathbf{2} = \mathbf{0} \otimes \mathbf{2} = \mathbf{4}) \oplus \exists n' \exists p' (\mathbf{2} = (s\ n') \otimes \mathbf{4} = (s\ p') \otimes P\ n'\ \mathbf{2}\ p')}\ \oplus}{\dfrac{\vdash plus\ \mathbf{2}\ \mathbf{2}\ \mathbf{4}}{\vdash \exists p.plus\ \mathbf{2}\ \mathbf{2}\ p}\ \exists}\ \mu$$

To complete this proof, we must construct a similar subproof verifying that $1+2=3$*. In particular, the witness used to instantiate the final* $\exists p$ *is, in fact, that sum. Unfortunately, proof construction in this system does not help us construct this sum's value. Instead, the first step in building such a proof bottom-up starts with guessing a value and checking that it is the correct sum.*

Example 4. *Given the definition of addition on natural numbers above, the following totality and determinancy formulas*

$$[\forall x_1 \forall x_2.\ nat\, x_1 \supset nat\, x_2 \supset \exists y.(plus(x_1,x_2,y) \wedge nat\, y)]$$
$$[\forall x_1 \forall x_2.\ nat\, x_1 \supset nat\, x_2 \supset \forall y_1 \forall y_2.\ plus(x_1,x_2,y_1) \supset plus(x_1,x_2,y_2) \supset y_1 = y_2]$$

can be proved in $\bar{\bar{\mu}}$*MALL where these formulas are polarized using the multiplicative connectives. These proofs require both induction and the* $\mu\nu$ *rule. Using the cut rule with (the obvious) proofs of* nat $\mathbf{2}$ *and* nat $\mathbf{3}$*, we know that* $\lambda y.(plus\ \mathbf{2}\ \mathbf{3}\ y)$ *denotes a singleton. In order to compute the sole member of the singleton* $\lambda y.(plus\ \mathbf{2}\ \mathbf{3}\ y)$*, we could perform cut-elimination with the inductively proved totality theorem in this example. Instead of such a* proof-reduction *approach to computation, the* proof search *approach starts by replacing the goal* $\exists y.(plus\ \mathbf{2}\ \mathbf{3}\ y \wedge nat\, y)$ *with* $\forall y.(plus\ \mathbf{2}\ \mathbf{3}\ y \supset nat\, y)$*. Attempting to prove this second formula leads to an incremental construction of the answer substitution for y, namely,* $\mathbf{5}$*.*

Assume that P is a purely positive predicate expression of type $i \to o$ and that we have a $\bar{\bar{\mu}}$MALL proof that P is a singleton. As we stated above, this means that we have a $\bar{\bar{\mu}}$MALL proof of $\forall x[Px \supset nat\, x]$. This proof can be understood as a means to compute the unique element of P except that the induction rule might be used in the proof of $\forall x[Px \supset nat\,x]$. Suppose we can force, however, the proof of this latter formula to be restricted so that the only form of induction is unfolding. In that case, such a restricted proof can provide an explicit computation. As the following example shows, it is not the case that if there is a $\bar{\mu}$MALL proof of $\forall x[Px \supset nat\,x]$ then it has a proof with the induction rule replaced by unfolding.

Example 5. *Let P be $\mu(\lambda R \lambda x. x = \mathbf{0} \oplus (R\ (s\ x)))$. Clearly, P denotes the singleton set containing zero. There is also a $\bar{\mu}$MALL proof that $\forall x[Px \supset nat\,x]$, but there is no (cut-free) proof of this theorem that uses unfolding instead of the more general induction rule: just using unfoldings leads to an unbounded proof search attempt which roughly follows the following outline.*

$$\dfrac{\vdash nat\ \mathbf{0} \qquad \dfrac{\begin{array}{c}\vdots\\ \vdash \overline{P\ (s\ (s\ y))}, nat\ y\end{array}}{\vdash \overline{P\ (s\ y)}, nat\ y}\ unfold, \&, \neq}{\vdash \overline{P\ y}, nat\ y}\ unfold, \&, \neq$$

Although proof search can contain potentially unbounded branches, we can still use the proof search concepts of unification and non-deterministic search to compute the value within a singleton. We define a non-deterministic algorithm as follows. The *state* of this algorithm is a triple of the form

$$\langle x_1, \ldots, x_n\,;\ B_1, \ldots, B_m\,;\ t \rangle,$$

where t is a term, $B_1, \ldots, B_m$ is a multiset of purely positive formulas, and all variables free in t and in the formulas $B_1, \ldots, B_m$ are in the set of variables $x_1, \ldots, x_n$. A *success state* is one of the form $\langle \cdot;\ \cdot;\ t \rangle$ (that is, when $n = m = 0$): such a state is said to have *value* t.

Given the state $S = \langle \Sigma;\ B_1, \ldots, B_m\,;\ t \rangle$ with $m \geq 1$, we can non-deterministically select one of the B_i formulas: for the sake of simplicity, assume that we have selected B_1. We define the transition $S \Rightarrow S'$ of state S to state S' by a case analysis of the top-level structure of B_1.

- If B_1 is $u = v$ and the terms u and v are unifiable with most general unifier θ, then we transition to $\langle \Sigma\theta\,;\ B_2\theta, \ldots, B_m\theta\,;\ t\theta \rangle$.
- If B_1 is $B \otimes B'$ then we transition to $\langle \Sigma;\ B, B', B_2, \ldots, B_m\,;\ t \rangle$.
- If B_1 is $B \oplus B'$ then we transition to either $\langle \Sigma;\ B, B_2, \ldots, B_m\,;\ t \rangle$ or $\langle \Sigma;\ B', B_2, \ldots, B_m\,;\ t \rangle$.
- If B_1 is $\mu B \vec{t}$ then we transition to $\langle \Sigma;\ B(\mu B)\vec{t}, B_2, \ldots, B_m\,;\ t \rangle$.
- If B_1 is $\exists y.\ B\ y$ then we transition to $\langle \Sigma, y;\ B\ y, B_2, \ldots, B_m\,;\ t \rangle$ assuming that y is not in Σ.

This non-deterministic algorithm is essentially applying left-introduction rules in a bottom-up fashion and, if there are two premises, selecting (non-deterministically) just one premise to follow.

Lemma 1. *Assume that P is a purely positive expression of type $i \to o$ and that $\exists y.Py$ has a $\bar{\mu}$LKp proof. There is a sequence of transitions from the initial state $\langle y;\ P\ y;\ y \rangle$ to a success state with value t such that $P\ t$ has a $\bar{\mu}$MALL proof.*

Proof. An *augmented state* is a structure of the form $\langle \Sigma \mid \theta\,;\ B_1 \mid \Xi_1, \ldots, B_m \mid \Xi_m\,;\ t \rangle$, where

- θ is a substitution with domain equal to Σ and which has no free variables in its range, and
- for all $i \in \{1, \ldots, m\}$, Ξ_i is a $\bar{\mu}$MALL proof of $\theta(B_i)$.

Clearly, if we strike out the augmented items (in red), we are left with a regular state. Given that we have a $\bar{\bar{\mu}}$LKp proof of $\exists y.Py$, conservativity (Theorem 3) ensures us that we have a $\bar{\bar{\mu}}$MALL proof of $\exists y.Py$. Thus, we there exists a $\bar{\bar{\mu}}$MALL proof Ξ_0 of $P\ t$ for some term t. Note that there is no occurrence of induction in Ξ_0. We now set the initial augmented state to $\langle y \mid [y \mapsto t];\ Py \mid \Xi_0;\ y\rangle$. As we detail now, the proof structures Ξ_i provide oracles that steer this non-deterministic algorithm to a success state with value t. Given the augmented state $\langle \Sigma \mid \theta;\ B_1 \mid \Xi_1, \ldots, B_m \mid \Xi_m;\ s\rangle$, we consider selecting the first pair $B_1 \mid \Xi_1$ and consider the structure of B_1.

- If B_1 is $B' \otimes B''$ then the last inference rule of Ξ_1 is $\otimes$ with premises Ξ' and Ξ'', and we make a transition to $\langle \Sigma \mid \theta;\ B' \mid \Xi', B'' \mid \Xi'', \ldots, B_m \mid \Xi_m;\ s\rangle$.
- If B_1 is $B' \oplus B''$ then the last inference rule of Ξ_1 is $\oplus$ and that rule selects either the first or the second disjunct. In either case, let Ξ' be the proof of its premise. Depending on which of these disjuncts is selected, we make a transition to either $\langle \Sigma \mid \theta;\ B' \mid \Xi', B_2 \mid \Xi_2, \ldots, B_m \mid \Xi_m;\ s\rangle$ or $\langle \Sigma \mid \theta;\ B'' \mid \Xi', B_2 \mid \Xi_2, \ldots, B_m \mid \Xi_m;\ s\rangle$, respectively.
- If B_1 is $\mu B\vec{t}$ then the last inference rule of Ξ_1 is μ. Let Ξ' be the proof of the premise of that inference rule. We make a transition to $\langle \Sigma \mid \theta;\ B(\mu B)\vec{t} \mid \Xi', B_2 \mid \Xi_2, \ldots, B_m \mid \Xi_m;\ s\rangle$.
- If B_1 is $\exists y.\ B\ y$ then the last inference rule of Ξ_1 is $\exists$. Let r be the substitution term used to introduce this $\exists$ quantifier and let Ξ' be the proof of the premise of that inference rule. Then we make a transition to $\langle \Sigma, w \mid \theta \circ \varphi;\ B\ w \mid \Xi', B_2 \mid \Xi_2, \ldots, B_m \mid \Xi_m;\ s\rangle$, where w is a variable not in Σ and φ is the substitution $[w \mapsto r]$. Here, we assume that the composition of substitutions satisfies the equation $(\theta \circ \varphi)(x) = \varphi(\theta(x))$.
- If B_1 is $u = v$ and the terms u and v are unifiable with most general unifier φ, then we make a transition to $\langle \Sigma\varphi \mid \rho;\ \varphi(B_2) \mid \Xi_2, \ldots, \varphi(B_m) \mid \Xi_m;\ (\varphi t)\rangle$ where ρ is the substitution such that $\theta = \varphi \circ \rho$.

In each of these cases, we must show that the transition is made to an augmented state. This is easy to show in all but the last two rules above. In the case of the transition due to $\exists$, we know that Ξ' is a proof of $\theta(B\ r)$, but that formula is simply $\varphi(\theta(B\ w))$ since w is new and r contains no variables free in Σ. In the case of the transition due to equality, we know that Ξ_1 is a proof of the formula $\theta(u = v)$ which means that θu and θv are the same terms and, hence, that u and v are unifiable and that θ is a unifier. Let φ be the most general unifier of u and v. Thus, there is a substitution ρ such that $\theta = \varphi \circ \rho$ and, for $i \in \{2, \ldots, m\}$, Ξ_i is a proof of $(\varphi \circ \rho)(B_i)$. Finally, termination of this algorithm is ensured since the number of occurrences of inference rules in the included proofs decreases at every step of the transition. Since we have shown that there is an augmented path that terminates, we have that there exists a path of states to a success state with value t. □

This lemma ensures that our search algorithm can compute a member from a non-empty set, give a $\bar{\bar{\mu}}$LKp proof that that set is non-empty.

We can now prove the following theorem about singleton sets. We abbreviate $(\exists x.P\,x) \wedge (\forall x_1 \forall x_2.P\,x_1 \supset P\,x_2 \supset x_1 = x_2)$ by $\exists! x.P\,x$ in the following theorem.

Theorem 5. *Assume that P is a purely positive expression of type $i \to o$ and that $\exists! y.Py$ has a $\bar{\bar{\mu}}$LKp proof. There is a sequence of transitions from the initial state $\langle y;\ P\ y;\ y\rangle$ to a success state of value t if and only if $P\ t$ has a $\bar{\bar{\mu}}$LKp proof.*

Proof. Given a (cut-free) $\bar{\bar{\mu}}$LKp proof of $\exists! y.Py$, that proof contains a $\bar{\bar{\mu}}$LKp proof of $\exists y.Py$. Since this formula is purely positive, there is a $\bar{\bar{\mu}}$MALL proof for $\exists y.Py$. The forward direction is immediate: given a sequence of transitions from the initial state $\langle y;\ P\ y;\ y\rangle$ to the success state $\langle\cdot;\ \cdot;\ t\rangle$, it is easy to build a $\bar{\bar{\mu}}$MALL proof of $P\ t$. Conversely, assume that there is a $\bar{\bar{\mu}}$LKp proof of $P\ t$ for some term t. By conservativity, there is a $\bar{\bar{\mu}}$MALL proof of $P\ t$ and, hence, of $\exists y.P\ y$. By Lemma 1, there is a sequence of transitions from initial state $\langle y;\ P\ y;\ y\rangle$ to the success state $\langle\cdot;\ \cdot;\ s\rangle$, where $P\ s$ has a $\bar{\bar{\mu}}$MALL proof. Given that Pt and Ps and $\forall x_1 \forall x_2.P\ x_1 \supset P\ x_2 \supset x_1 = x_2$ all have $\bar{\bar{\mu}}$LKp$^+$ proofs, using the cut rule, we can conclude that $t = s$. □

Thus, a (naive) proof-search algorithm involving both unification and non-deterministic search is sufficient for computing the functions encoded in relations.

While it is easy to encode the proof of totality for the Ackermann function in $\bar{\bar{\mu}}$LKp, it seems unlikely that a totality proof for that function can be done within $\bar{\bar{\mu}}$MALL. This separation between $\bar{\bar{\mu}}$LKp and $\bar{\bar{\mu}}$MALL was conjectured by Baelde [1, Section 3.5]. There are also several other linear logic style systems for which the totality of Ackermann's function is known to be not provable. In particular, if we developed a Curry-Howard interpretation of $\bar{\bar{\mu}}$MALL, it would yield a system close to the linear λ-terms $H(\emptyset)$ of [14], which is known to capture exactly primitive recursive functions (see also similar results in [13]).

Acknowledgment: We thank the anonymous reviewers of an earlier draft of this paper for they valuable comments.

A Consistency of $\bar{\bar{\mu}}$LKp

By second-order linear logic, *LL*2, we mean the logic of MALL with the addition of the following logical connectives: two exponentials ! and ?, negation $(\cdot)^\perp$, equality and non-equality, and first-order and second-order quantification (no occurrences of fixed points are permitted). Cut-elimination of this version of *LL*2 follows from Girard's original cut-elimination proof [9] (see also [15]) and the cut-elimination proofs known for equality and non-equality [11, 16].

We translate $\bar{\bar{\mu}}$LKp formulas into *LL*2 formulas by translating fixed point expressions into second-order quantified formulas. The least fixed point expression $\mu B\vec{x}$ should be translated to a formula roughly of the form $\forall S\Big(\ !(\forall\vec{y}\ .\ BS\vec{y} \multimap S\vec{y}) \multimap S\vec{x}\ \Big)$. This translation must also insert ? into formulas in order to account for the fact that in $\bar{\bar{\mu}}$LKp, any formula can be contracted and weakened at any point in a proof. The translation is given as follows.

- $\lceil t = s\rceil = ?(t = s)$ and $\lceil t \neq s\rceil = ?(t \neq s)$
- $\lceil \forall x.Px\rceil = ?\forall x.\lceil Px\rceil$ and $\lceil \exists x.Px\rceil = ?\exists x.\lceil Px\rceil$.
- $\lceil B \otimes C\rceil = ?(\lceil B\rceil \otimes \lceil C\rceil)$, $\lceil B \parr C\rceil = ?(\lceil B\rceil \parr \lceil C\rceil)$, $\lceil B \,\&\, C\rceil = ?(\lceil B\rceil \,\&\, \lceil C\rceil)$,
  $\lceil B \oplus C\rceil = ?(\lceil B\rceil \oplus \lceil C\rceil)$
- $\lceil 1\rceil = ?1$, $\lceil \perp\rceil = ?\perp$, $\lceil 0\rceil = ?0$, $\lceil \top\rceil = ?\top$
- $\lceil \mu B\vec{x}\rceil = ?\forall S[(?\exists\vec{y}\ .\ \lceil B\rceil S\vec{y} \otimes (S\vec{y})^\perp) \parr S\vec{x}\ ]$ and $\lceil \nu B\vec{x}\rceil = ?\exists S[(!\forall\vec{y}\ .\ \lceil B\rceil S\vec{y} \parr (S\vec{y})^\perp) \otimes S\vec{x}\ ]$
- $\lceil A\rceil = A$ where A is an atomic formula.
- The $\lceil\cdot\rceil$ operator commutes with λ-abstraction: $\lceil \lambda x.B\rceil = \lambda x.\lceil B\rceil$. This feature of $\lceil\cdot\rceil$ permit translating invariants and the body of fixed point expressions.
- The $\lceil\cdot\rceil$ operator can be applied to a multiset of formulas: $\lceil\Gamma\rceil = \{\lceil P\rceil \mid P \in \Gamma\}$.

Note that when B is the λ-abstraction $\lambda p \lambda \vec{x}.C$, where C is a $\bar{\bar{\mu}}$MALL formula, p is a first-order predicate variable, and $\vec{x}$ is a list of first-order variables, then $\lceil B \rceil \lceil S \rceil \vec{t}$ is equal to $\lceil BS\vec{t} \rceil$ up to λ-conversion. We shall also need the following inference rule in $LL2$, which is a kind of generalization of the cut rule.

$$\dfrac{\vdash \Gamma, BQ\vec{t} \quad \vdash \neg(Q\vec{x}), P\vec{x}}{\vdash \Gamma, BP\vec{t}}\; deep.$$

Here, of course, the first-order variables $\vec{x}$ are new. Also, the expression B has the type that takes a first-order predicate to a first-order predicate and also *monotonic*, meaning that there are no occurrences of negated predicate variables in B. It is proved in [3, Proposition 2] that this rule is admissible in $LL2$. This rule essentially allows us to move from the fact that $Q \subseteq P$ and to the fact that $BQ \subseteq BP$.

Lemma 2. *If* $\vdash \Gamma$ *is derivable in* $\bar{\bar{\mu}}$*LKp then* $\vdash \lceil \Gamma \rceil$ *is derivable in* $LL2$.

Proof. We proceed by induction on the structure of cut-free $\bar{\bar{\mu}}$LKp proofs. In particular, assume that $\vdash \Gamma$ has a cut-free $\bar{\bar{\mu}}$LKp proof Ξ.

Case: The last inference rule of Ξ comes from Figure 1, *i.e.*, it is an introduction rules for a propositional connective, a unit, or a quantifier. For example, assume that this last inference rule is the following $\otimes$ introduction rule.

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

By the inductive assumption, $\vdash \lceil \Gamma \rceil, \lceil P \rceil$ and $\vdash \lceil \Delta \rceil, \lceil Q \rceil$ have $LL2$ proofs. Hence, $\vdash \lceil \Gamma \rceil, \lceil \Delta \rceil, \lceil P \rceil \otimes \lceil Q \rceil$ has an $LL2$ proof. By using the dereliction rule for ? and the definition of $\lceil \cdot \rceil$, we know that $\vdash \lceil \Gamma, \Delta \rceil, \lceil P \otimes Q \rceil$ has an $LL2$ proof.

Case: The last rule is weakening W or contraction C. Since the image of $\lceil \cdot \rceil$ always has a ? exponential as its top-level connective, the corresponding $LL2$ inference rule is built with the same structural rule.

Case: The last inference rule of Ξ is one of the fixed point rules from Figure 1. Assume it is the $\mu\nu$ rule. The desired translation of this inference rule into $LL2$ is

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash \lceil B \rceil S\vec{y}, \lceil \overline{B} \rceil (\lambda \vec{w} (S\vec{w})^{\perp})\vec{y} \quad \dfrac{}{\vdash (S\vec{y})^{\perp}, \neg((S\vec{x})^{\perp})}\; init}{\vdash \lceil B \rceil S\vec{y} \otimes (S\vec{y})^{\perp}, \lceil \overline{B} \rceil (\lambda \vec{w} (S\vec{w})^{\perp})\vec{y} \parr \neg((S\vec{y})^{\perp})}\; \parr, \otimes}{\vdash ?\exists \vec{y}. \lceil B \rceil S\vec{y} \otimes (S\vec{y})^{\perp}, !(\forall \vec{y}. \lceil \overline{B} \rceil (\lambda \vec{w} (S\vec{w})^{\perp})\vec{y} \parr \neg((S\vec{y})^{\perp}))}\; !R, ?D, \forall, \exists \quad \dfrac{}{\vdash S\vec{x}, (S\vec{x})^{\perp}}\; init}{\dfrac{\vdash ?(\exists \vec{y}. \lceil B \rceil S\vec{y} \otimes (S\vec{y})^{\perp}) \parr S\vec{x}\, , !(\forall \vec{y}. \lceil \overline{B} \rceil (\lambda \vec{w} (S\vec{w})^{\perp})\vec{y} \parr \neg((S\vec{y})^{\perp})) \otimes (S\vec{x})^{\perp}}{\vdash ?(\exists \vec{y}\, .\, \lceil B \rceil S\vec{y} \otimes (S\vec{y})^{\perp}) \parr S\vec{x}\, , \exists S[!(\forall \vec{y}\, .\, \lceil \overline{B} \rceil S\vec{y} \parr (S\vec{y})^{\perp}) \otimes S\vec{x}\,]}\; \exists \lambda \vec{w} (S\vec{w})^{\perp}}\; \parr, \otimes}{\vdash ?\forall S[?(\exists \vec{y}\, .\, \lceil B \rceil S\vec{y} \otimes (S\vec{y})^{\perp}) \parr S\vec{x}\,], ?\exists S[!(\forall \vec{y}\, .\, \lceil \overline{B} \rceil S\vec{y} \parr (S\vec{y})^{\perp}) \otimes S\vec{x}\,]}\; ?D, \forall$$

An induction on the structure of B provides a proof that there is an $LL2$ proof of remaining open premise.

Assume instead that the last rule of Ξ is the introduction for ν, namely,

$$\dfrac{\vdash \Gamma, S\vec{t} \quad \vdash BS\vec{x}, \overline{(S\vec{x})}}{\vdash \Gamma, \nu B\vec{t}}\; \nu.$$

The higher-order quantifier that appears in the $LL2$ encoding is instantiated with $\lceil S \rceil$. Thus, the desired $LL2$ proof is

$$\dfrac{\dfrac{\dfrac{\vdash \lceil BS\vec{x} \rceil, \lceil \overline{S\vec{x}} \rceil \quad \vdash \neg(\lceil \overline{S\vec{x}} \rceil), \neg(\lceil S \rceil \vec{x})}{\vdash \lceil B \rceil \lceil S \rceil \vec{x}, \neg(\lceil S \rceil \vec{x})}\; cut \quad \vdash \lceil \Gamma \rceil, \lceil S\vec{t} \rceil}{\vdash \lceil \Gamma \rceil, !(\forall \vec{y}\, .\, \lceil B \rceil \lceil S \rceil \vec{y} \parr \neg(\lceil S \rceil \vec{y})) \otimes \lceil S \rceil \vec{t}}\; \otimes, \forall, \parr}{\vdash \lceil \Gamma \rceil, ?\exists S[!(\forall \vec{y}\, .\, \lceil B \rceil S\vec{y} \parr \neg(S\vec{y})) \otimes S\vec{t}\,]}\; ?D, \exists\, S \mapsto \lceil S \rceil$$

By the inductive hypothesis, the leftmost and rightmost premises have *LL*2 proof. Induction on first-order abstractions such as S shows that the middle premise also has an *LL*2 proof.

Assume instead that the last rule of Ξ is the introduction for μ, namely,

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

We first show that $\vdash \lceil B\rceil\lceil \mu B\rceil\vec{t} \multimap \lceil \mu B\vec{t}\rceil$ has an *LL*2 proof for all B and $\vec{t}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \overline{\lceil B\rceil\lceil \mu B\rceil\vec{t}}, \lceil B\rceil\lceil \mu B\rceil\vec{t}}\ \mathit{init} \qquad \begin{array}{c}\Xi\\ \vdash ?(\exists\vec{y}\,.\,\lceil B\rceil S\vec{y}\otimes(S\vec{y})^{\perp}), \overline{\lceil \mu B\rceil\vec{x}}, S\vec{x}\end{array}}{\vdash \overline{\lceil B\rceil\lceil \mu B\rceil\vec{t}}, ?(\exists\vec{y}\,.\,\lceil B\rceil S\vec{y}\otimes(S\vec{y})^{\perp}), \lceil B\rceil S\vec{t}}\ \mathit{deep} \qquad \dfrac{}{\vdash (S\vec{t})^{\perp}, S\vec{t}}\ \mathit{init}}{\vdash \overline{\lceil B\rceil\lceil \mu B\rceil\vec{t}}, ?(\exists\vec{y}\,.\,\lceil B\rceil S\vec{y}\otimes(S\vec{y})^{\perp}), \lceil B\rceil S\vec{t}\otimes(S\vec{t})^{\perp}, S\vec{t}}\ \otimes}{\vdash \overline{\lceil B\rceil\lceil \mu B\rceil\vec{t}}, ?(\exists\vec{y}\,.\,\lceil B\rceil S\vec{y}\otimes(S\vec{y})^{\perp}), S\vec{t}}\ C,D,\exists}{\vdash \overline{\lceil B\rceil\lceil \mu B\rceil\vec{t}}, \lceil \mu B\vec{t}\rceil}\ ?,\forall,\parr$$

Here, Ξ is a straightforward *LL*2 proof. Finally, using this proof of $\vdash \overline{\lceil B\rceil\lceil \mu B\rceil\vec{t}}, \lceil \mu B\vec{t}\rceil$ and the cut rule for *LL*2, we have shown the soundness of the μ rule in Figure 1. □

Proof of Theorem 1. Assume that $\vdash B$ and $\vdash \overline{B}$ have $\bar{\bar{\mu}}$LKp proofs. By Lemma 2, we know that $\vdash \lceil B\rceil$ and $\vdash \lceil\overline{B}\rceil$ have *LL*2 proofs. While it is not the case that $\lceil\overline{B}\rceil = (\lceil B\rceil)^{\perp}$, a simple induction on the structure of B shows that $\lceil\overline{B}\rceil \vdash (\lceil B\rceil)^{\perp}$ is provable in *LL*2. Since *LL*2 has a cut rule, we know that there is an *LL*2 proof of $\vdash \cdot$ (the empty sequent). By the cut-elimination theorem of *LL*2, this sequent also has a cut-free *LL*2 proof, which is impossible. □

B $\bar{\bar{\mu}}$LKp(Σ_1) is conservative over $\bar{\bar{\mu}}$MALL for Π_2-formulas

In this section we prove that any Π_2 formula provable in $\bar{\bar{\mu}}$LKp(Σ_1) is provable in $\bar{\bar{\mu}}$MALL. This conservativity result can be applied to the formulas stating the *totality* and *determinancy* properties (see Section 5) of relations defined by Σ_1-formulas, since they are all Π_2 formulas. The proof of this result would be aided greatly if we had a focusing theorem for $\bar{\bar{\mu}}$LKp. If we take the focused proof system for $\bar{\bar{\mu}}$MALL given in [2, 3] and add contraction and weakening in the usual fashion, we have a natural candidate for a focused proof system for $\bar{\bar{\mu}}$LKp. However, the completeness of that proof system is currently open. As Girard points out in [10], the completeness of such a focused (cut-free) proof system would allow the extraction of the constructive content of classical Π^0_2 theorems, and we should not expect such a result to follow from the usual ways that we prove cut-elimination and the completeness of focusing. As a result of not possessing such a focused proof system for $\bar{\bar{\mu}}$LKp, we must reproduce aspects of focusing in order to prove our conservation result.

Definition 1. *A* reduced sequent *is a sequent that contains only purely negative, purely positive, and* Π_2 *formulas. If* Γ_1 *and* Γ_2 *are reduced sequents, we say that* Γ_1 contains Γ_2 *if* Γ_2 *is a sub-multiset of* Γ_1. *Finally, we say that a reduced sequent is a* pointed *sequent if it contains exactly one formula that is either purely positive or* Π_2.

Definition 2. *A* positive region *is a cut-free* $\bar{\bar{\mu}}$*LKp*(Σ_1) *proof that contains only the inference rules* $\mu\nu$, *contractions, weakening, and introduction rules for the positive connectives.*

Definition 3. *The $C\nu\nu$ rule is the following derived rule of inference.*

$$\frac{\vdash \Gamma, S\vec{t}, U\vec{t} \qquad \vdash BU\vec{x}, \overline{U\vec{x}} \qquad \vdash BS\vec{x}, \overline{S\vec{x}}}{\vdash \Gamma, \nu B\vec{t}}\; C\nu\nu$$

The $C\nu\nu$ rule is justified as the following combination of ν and contraction rules.

$$\frac{\dfrac{\dfrac{\vdash \Gamma, S\vec{t}, U\vec{t} \qquad \vdash BU\vec{x}, \overline{U\vec{x}}}{\vdash \Gamma, \nu B\vec{t}, S\vec{t}}\;\nu \qquad \vdash BS\vec{x}, \overline{S\vec{x}}}{\vdash \Gamma, \nu B\vec{t}, \nu B\vec{t}}\;\nu}{\vdash \Gamma, \nu B\vec{t}}\; C$$

Since we are working within $\bar{\bar{\mu}}$LKp(Σ_1), the invariants S and U are purely positive.

Definition 4. *A* negative region *is a cut-free $\bar{\bar{\mu}}LKp(\Sigma_1)$ partial proof in which the open premises are all reduced sequent and where the only inference rules are introductions for negative connectives plus the $C\nu\nu$ rule.*

Lemma 3. *If a reduced sequent Γ has a positive region proof then Γ contains a pointed sequent that has a $\bar{\bar{\mu}}$MALL proof.*

Proof. This proof is a simple generalization of the proof of Theorem 3. □

Lemma 4. *If every premise of a negative region contains a pointed sequent with a $\bar{\bar{\mu}}$MALL proof, then the conclusion of the negative region contains a pointed sequent with a $\bar{\bar{\mu}}$MALL proof.*

Proof. This proof is by induction on the height of the negative region. The most interesting case to examine is the one where the last inference rule of the negative region is the $C\nu\nu$ rule. Referring to the inference rule displayed above, the inductive hypothesis ensures that the reduced sequent $\vdash \Gamma, S\vec{t}, U\vec{t}$ contains a pointed sequent Δ, C where Δ is a multiset of purely negative formula in Γ and where the formula C (that is either purely positive or is Π_2) is either a member of Γ or is equal to either $S\vec{t}$ or $U\vec{t}$. In the first case, Δ, C is also contained in the endsequent $\Gamma, \nu B\vec{t}$. In the second case, we have one of the following proofs:

$$\frac{\vdash \Delta, S\vec{t} \quad \vdash BS\vec{x}, \overline{S\vec{x}}}{\vdash \Gamma, \nu B\vec{t}}\;\nu \qquad\qquad \frac{\vdash \Delta, U\vec{t} \quad \vdash BU\vec{x}, \overline{U\vec{x}}}{\vdash \Gamma, \nu B\vec{t}}\;\nu$$

depending on whether or not C is $S\vec{t}$ or $U\vec{t}$. □

Lemma 5. *If the reduced sequent Γ has a cut-free $\bar{\bar{\mu}}LKp(\Sigma_1)$ proof then Γ has a proof that can be divided into a negative region that proves Γ in which all its premises have positive region proofs.*

Proof. This lemma is proved by appealing to the permutation of inference rules. As shown in [2], the introduction rules for negative connectives permute down over all inference rules in $\bar{\bar{\mu}}$MALL. Not considered in that paper is how such negative introduction rules permute down over contractions. It is easy to check that such permutations do, in fact, happen except in the case of the ν rule. In general, contractions below a ν rule will not permute upwards, and, as a result, the negative region is designed to include the $C\nu\nu$ rule (where contraction is stuck with the ν rule). As a result, negative rules (including $C\nu\nu$) permute down while contraction and introductions of positive connectives permute upward. This gives rise to the two-region proof structure. □

By combining the results of this section we have a proof of Theorem 4.

References

[1] David Baelde (2008): *A linear approach to the proof-theory of least and greatest fixed points*. Ph.D. thesis, Ecole Polytechnique. Available at `http://arxiv.org/abs/0910.3383`.

[2] David Baelde (2012): *Least and greatest fixed points in linear logic*. *ACM Trans. on Computational Logic* 13(1), pp. 2:1–2:44, doi:10.1145/2071368.2071370.

[3] David Baelde & Dale Miller (2007): *Least and greatest fixed points in linear logic*. In N. Dershowitz & A. Voronkov, editors: *International Conference on Logic for Programming and Automated Reasoning (LPAR)*, *LNCS* 4790, pp. 92–106, doi:10.1007/978-3-540-75560-9_9.

[4] James Brotherston & Alex Simpson (2011): *Sequent calculi for induction and infinite descent*. *J. of Logic and Computation* 21(6), pp. 1177–1216, doi:10.1093/logcom/exq052.

[5] Alonzo Church (1940): *A Formulation of the Simple Theory of Types*. *J. of Symbolic Logic* 5, pp. 56–68, doi:10.2307/2266170.

[6] Anupam Das (2020): *On the logical complexity of cyclic arithmetic*. *Log. Methods Comput. Sci* 16(1), doi:10.4230/LIPIcs.FSCD.2021.29.

[7] Thomas Ehrhard & Farzad Jafarrahmani (2021): *Categorical models of Linear Logic with fixed points of formulas*. In: *36th ACM/IEEE Symposium on Logic in Computer Science (LICS 2021), Jun 2021, Rome, Italy*, IEEE, pp. 1–13, doi:10.1109/LICS52264.2021.9470664.

[8] Harvey M. Friedman (1978): *Classically and Intuitionistically Provably Recursive Functions*. In G. H. Müller & Dana S. Scott, editors: *Higher Order Set Theory*, Springer Verlag, Berlin, pp. 21–27, doi:10.1007/BFb0103100.

[9] Jean-Yves Girard (1987): *Linear Logic*. *Theoretical Computer Science* 50(1), pp. 1–102, doi:10.1016/0304-3975(87)90045-4.

[10] Jean-Yves Girard (1991): *A new constructive logic: classical logic*. *Math. Structures in Comp. Science* 1, pp. 255–296, doi:10.1017/S0960129500001328.

[11] Jean-Yves Girard (1992): *A Fixpoint Theorem in Linear Logic*. An email posting archived at `https://www.seas.upenn.edu/~sweirich/types/archive/1992/msg00030.html` to the linear@cs.stanford.edu mailing list.

[12] William A. Howard (1980): *The formulae-as-type notion of construction, 1969*. In J. P. Seldin & R. Hindley, editors: *To H. B. Curry: Essays in Combinatory Logic, Lambda Calculus, and Formalism*, Academic Press, New York, pp. 479–490.

[13] Denis Kuperberg, Laureline Pinault & Damien Pous (2021): *Cyclic proofs, System T, and the power of contraction*. In: *Proc. of the ACM on Programming Languages - POPL*, 5, pp. 1–28, doi:10.1145/3434282.

[14] Ugo Dal Lago (2009): *The geometry of linear higher-order recursion*. *ACM Trans. Comput. Log.* 10(2), pp. 8:1–8:38, doi:10.1145/1462179.1462180.

[15] Mitsuhiro Okada (1999): *Phase Semantic Cut-Elimination and Normalization Proofs of First- and Higher-Order Linear Logic*. *Theor. Comput. Sci* 227(1-2), pp. 333–396, doi:10.1016/S0304-3975(99)00058-4.

[16] Peter Schroeder-Heister (1993): *Rules of Definitional Reflection*. In M. Vardi, editor: *8th Symp. on Logic in Computer Science*, IEEE Computer Society Press, IEEE, pp. 222–232, doi:10.1109/LICS.1993.287585.

[17] Alex Simpson (2017): *Cyclic Arithmetic Is Equivalent to Peano Arithmetic*. In Javier Esparza & Andrzej S. Murawski, editors: *Foundations of Software Science and Computation Structures - 20th International Conference, FoSSaCS*, *LNCS* 10203, pp. 283–300, doi:10.1007/978-3-662-54458-7_17.

[18] Sten-Ake Tärnlund (1977): *Horn Clause Computability*. *BIT* 17, pp. 215–226.