Fundamenta Informaticae, Volume 194, Issue 2, Article 2, 2025
DOI: 10.46298/fi.12736
https://fi.episciences.org/

Peano Arithmetic and μMALL

Matteo Manighetti
Inria Saclay
Saclay, France

Dale Miller
Inria-Saclay and LIX, Institut Polytechnique de Paris
Palaiseau, France

Abstract. Formal theories of arithmetic have traditionally been based on either classical or intuitionistic logic, leading to the development of Peano and Heyting arithmetic, respectively. We propose to use μMALL as a formal theory of arithmetic based on linear logic. This formal system is presented as a sequent calculus proof system that extends the standard proof system for multiplicative-additive linear logic (MALL) with the addition of the universal and existential quantifiers (first-order quantifiers), term equality and non-equality, and the least and greatest fixed point operators. We first demonstrate how functions defined using μMALL relational specifications can be computed using a simple proof search algorithm. By incorporating weakening and contraction into μMALL, we obtain $\mu\mathrm{LK}_p^+$, a natural candidate for a classical sequent calculus for arithmetic. While important proof theory results are still lacking for $\mu\mathrm{LK}_p^+$ (including cut-elimination and the completeness of focusing), we prove that $\mu\mathrm{LK}_p^+$ is consistent and that it contains Peano arithmetic. We also prove some conservativity results regarding $\mu\mathrm{LK}_p^+$ over μMALL.

1. Introduction

First-order logic formulas are built from propositional connectives, first-order quantifiers, first-order terms, and the class of non-logical constants called *predicates* that denote relations between terms. Moving from first-order logic to first-order arithmetic, one introduces induction principles and banishes the predicate constants by formally defining relations between terms using those inductive principles. When moving from classical logic to arithmetic in this fashion, one arrives at a presentation of Peano Arithmetic. In this paper, we continue the project of studying arithmetic based instead on linear logic, which was initiated in [2, 3, 7], and where this linearized version of arithmetic was called μMALL. Since that earlier work, many researchers (*e.g.* [16, 29]) have adopted this name to refer to the propositional fragment of this logic, *i.e.*, the fragment without first-order terms, quantification, and equality. To emphasize our focus on first-order structures, we will use the name $\bar{\bar{\mu}}$MALL (and $\bar{\bar{\mu}}\mathrm{LK}_p$) in the rest of this paper.

Linear logic has played various roles in computational logic. Many applications rely on the ability of linear logic to capture the multiset rewriting paradigm [9], which, in turn, can encode Petri nets [21], process calculi [32, 36], and stateful computations [28, 37]. Our use of linear logic here will have none of that flavor. While the sequent calculus we use to present $\bar{\bar{\mu}}$MALL in Section 3 is based on multisets of formulas, we shall not model computation as some rewriting of multisets of atomic-formulas-as-tokens. In contrast, when we use linear logic connectives within arithmetic, we capture computation and deduction via familiar means that rely on relations between numerical expressions. We propose linearized arithmetic not to build a non-standard arithmetic but to better understand computation and reasoning in arithmetic.

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 contrasts the uses of the propositional subset of $\bar{\bar{\mu}}$MALL as a typing system (see, for example, [19]). We shall limit ourselves to using invariants to reason inductively about fixed points instead of employing other methods, such as infinitary proof systems (e.g., [11]) and cyclic proof systems (e.g., [15, 42]).

We begin our analysis of arithmetic by demonstrating that functions defined relationally in $\bar{\bar{\mu}}$MALL can be directly computed from their specifications using unification and backtracking search (Section 4). We then introduce a new proof system, $\bar{\bar{\mu}}\mathrm{LK}_p^+$, which extends $\bar{\bar{\mu}}$MALL with the weakening and contraction rules. While the addition of these rules provides a natural foundation for classical logic, the precise nature of $\bar{\bar{\mu}}\mathrm{LK}_p^+$ is not well understood. In particular, we do not yet know if $\bar{\bar{\mu}}\mathrm{LK}_p$, the cut-free version of $\bar{\bar{\mu}}\mathrm{LK}_p^+$, is equivalent to $\bar{\bar{\mu}}\mathrm{LK}_p^+$ or if it admits a complete focusing proof system. In this paper, we establish the consistency of $\bar{\bar{\mu}}\mathrm{LK}_p^+$, demonstrate its capacity to encode Peano Arithmetic (Section 6), and prove specific conservativity results of $\bar{\bar{\mu}}\mathrm{LK}_p$ over $\bar{\bar{\mu}}$MALL (Section 7).

2. Terms and formulas

We use Church's approach [12] to define terms, formulas, and abstractions over these by making them all simply typed λ-terms. The primitive type o denotes formulas (of linear and classical logics). In this paper, we assume that there is a second primitive type ι and that the (ambient) signature contains the constructors $z\colon \iota$ (zero) and $s\colon \iota \to \iota$ (successor). We abbreviate the terms z, $(s\ z)$, $(s\ (s\ z))$, $(s\ (s\ (s\ z)))$, etc by **0**, **1**, **2**, **3**, etc.

2.1. Logical connectives involving type ι

We first present the logical connectives that relate to first-order structures. The two quantifiers $\forall$ and $\exists$ are both given the type $(\iota \to o) \to o$: the terms $\forall(\lambda x.B)$ and $\exists(\lambda x.B)$ of type o are abbreviated as $\forall x.B$ and $\exists x.B$, respectively. Equality $=$ and non-equality $\neq$ are both of the type $\iota \to \iota \to o$. For $n \geq 0$, the least fixed point operator of arity n is written as μ_n and the greatest fixed point operator of arity n is written as ν_n, and they both have the type $(A \to A) \to A$ where A is the type $\iota \to \cdots \to \iota \to o$ in which there are n occurrences of ι. We seldom write explicitly the arity of fixed points as it can usually be determined from context when its value is important. The pairs of connectives $\langle \forall, \exists \rangle$, $\langle \mu, \nu \rangle$, and $\langle =, \neq \rangle$ are De Morgan duals.

Our formalizations of arithmetic do not contain predicate constants: we do not admit any non-logical symbols of type $\iota \to \cdots \to \iota \to o$. Consequently, there are no atomic formulas in the traditional sense, *i.e.*, formulas headed by non-logical symbols. Equality, non-equality, and fixed-point operators are treated as logical connectives, as they will be given introduction rules in the sequent calculus proof systems we will soon introduce.

We shall use the usual rules for λ-conversion, namely, α, β, and η conversion [12], as equality on both terms and formulas. In general, we assume that terms and formulas are in β-normal form.

2.2. Propositional connectives of linear logic

The eight linear logic connectives for MALL are the following.

	conjunction	true	disjunction	false
multiplicative	$\otimes$	1	$⅋$	$\bot$
additive	$\&$	$\top$	$\oplus$	0

The four binary connectives have type $o \to o \to o$, and the four units have type o. (The use of 0 and 1 as logical connectives is unfortunate for a paper about arithmetic. As we mentioned above, numerals are written in boldface.) Formulas involving the set of logical connectives in Section 2.1 and these propositional connectives are called $\bar{\bar{\mu}}$MALL formulas, a logic that was first proposed in [7]. Many of the proof-theoretic properties of $\bar{\bar{\mu}}$MALL will be summarized in Section 3.

We do not treat negation as a logical connective: when B is a formula, we write $\overline{B}$ to denote the formula resulting from taking the De Morgan dual of B. We occasionally use the linear implication $B \multimap C$ as an abbreviation for $\overline{B} ⅋ C$. We also use this overline notation when B is the body of a fixed point expression, *i.e.*, when B has the form $\lambda p \lambda \vec{x}.C$ where C is a formula, p is a first-order predicate variable, and $\vec{x}$ is a list of first-order variables, then $\overline{B}$ is $\lambda p \lambda \vec{x}.\overline{C}$ [3, Definition 2.1]. For example, if B is $[\lambda p \lambda x.x = z \oplus \exists y.x = (s\ (s\ y)) \otimes p\ y]$ then $\overline{B}$ is $[\lambda p \lambda x.x \neq z \ \&\ \forall y.x \neq (s\ (s\ y)) ⅋ p\ y]$.

2.3. Polarized and unpolarized formulas

The connectives of linear logic are given a *polarity* as follows. The *negative* connectives are $⅋$, $\bot$, $\&$, $\top$, $\forall$, $\neq$, and ν while their De Morgan duals—namely, $\otimes$, 1, $\oplus$, 0, $\exists$, $=$, and μ—are positive. A $\bar{\bar{\mu}}$MALL formula is positive or negative depending only on the polarity of its topmost connective. The polarity flips between B and $\overline{B}$. We shall also call $\bar{\bar{\mu}}$MALL formulas *polarized formulas*.

Unpolarized formulas are built using the four usual classical propositional logic connectives $\wedge$, *tt*, $\vee$, *ff* plus $=$, $\neq$, $\forall$, $\exists$, μ, and ν. Thus, the six connectives with i in their typing can appear in polarized and unpolarized formulas. Unpolarized formulas are also called *classical logic formulas*. Note that unpolarized formulas do not contain negations. We shall extend the notation $\overline{B}$ to unpolarized formulas

B in the same sense as used with polarized formulas. For convenience, we will occasionally allow implications in unpolarized formulas: in those cases, we treat $P \supset Q$ as $\overline{P} \vee Q$.

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 *ff* 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 expressions, such as $((\mu\lambda P\lambda x(B\ P\ x))\ t)$, introduce variables of predicate type (here, P) into their scope. In the case of the μ fixed point, any formula built using that predicate variable as its topmost symbol will be considered positively polarized. Dually, if the ν operator is used instead, any formula built using the predicate variable it introduces is considered negatively polarized. For example, expression $(p\ y)$ in $[\mu\lambda p\lambda x.x = z \oplus \exists y.x = (s\ (s\ y)) \otimes p\ y]$ is polarized positively while that same expression in $[\nu\lambda p\lambda x.x \neq z\ \&\ \forall y.x \neq (s\ (s\ y))\ ⅋\ p\ y]$ is polarized negatively.

2.4. The polarization hierarchy

A formula is *purely positive* (resp., *purely negative*) if every logical connective it contains is positive (resp., negative). Taking inspiration from the familiar notion of the arithmetical hierarchy, we define the following collections of formulas. The formulas in $\mathbf{P}_1$ are the purely positive formulas, and the formulas in $\mathbf{N}_1$ are the purely negative formulas. More generally, for $n \geq 1$, the $\mathbf{N}_{n+1}$-formulas are those negative formulas for which every occurrence of a positive subformula is a $\mathbf{P}_n$-formula. Similarly, the $\mathbf{P}_{n+1}$-formulas are those positive formulas for which every occurrence of a negative subformula is an $\mathbf{N}_n$-formula. A formula in $\mathbf{P}_n$ or in $\mathbf{N}_n$ has at most $n-1$ alternations of polarity. Clearly, the dual of a $\mathbf{P}_n$-formula is an $\mathbf{N}_n$-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 $\mathbf{P}_n$ if B is a $\mathbf{P}_n$-formula.

Note that for all $n \geq 1$, if B is an unpolarized Π^0_n-formula (in the usual arithmetic hierarchy) then there is a polarized version of B that is $\mathbf{N}_n$. Similarly, if B is an unpolarized Σ^0_n-formula then there is a polarized version of B that is $\mathbf{P}_n$.

3. Linear and classical proof systems for polarized formulas

3.1. The $\bar{\bar{\mu}}$MALL and $\bar{\bar{\mu}}\mathbf{LK}^+_p$ proof systems

The $\bar{\bar{\mu}}$MALL proof system [3,7] 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 from appearing free in any formula in the conclusion of that rule. In the $\neq$-introduction rule, if the terms t and t' are not unifiable, the premise is empty and the conclusion follows immediately.

The choice of using Church's λ-notation provides an elegant treatment of higher-order substitutions (needed for handling 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 η-long, β-normal form

$$
\frac{\vdash \Gamma, B, C}{\vdash \Gamma, B \parr C}\ \parr
\qquad
\frac{\vdash \Gamma}{\vdash \Gamma, \bot}\ \bot
\qquad
\frac{\vdash \Gamma, B \quad \vdash \Delta, C}{\vdash \Gamma, \Delta, B \otimes C}\ \otimes
\qquad
\frac{}{\vdash 1}\ 1
$$

$$
\frac{\vdash \Gamma, B \quad \vdash \Gamma, C}{\vdash \Gamma, B \,\&\, C}\ \&
\qquad
\frac{}{\vdash \Gamma, \top}\ \top
\qquad
\frac{\vdash \Gamma, B_i}{\vdash \Gamma, B_0 \oplus B_1}\ \oplus
$$

$$
\frac{\{\ \vdash \Gamma\theta : \theta = mgu(t, t')\ \}}{\vdash \Gamma, t \neq t'}\ \neq
\qquad
\frac{}{\vdash t = t}\ =
\qquad
\frac{\vdash \Gamma, By}{\vdash \Gamma, \forall x.Bx}\ \forall
\qquad
\frac{\vdash \Gamma, Bt}{\vdash \Gamma, \exists x.Bx}\ \exists
$$

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

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

$$
\frac{\vdash \Gamma, B(\nu B)\vec{t}}{\vdash \Gamma, \nu B\vec{t}}\ \textit{unfold}
\qquad
\frac{}{\vdash B, \overline{B}}\ \textit{init}
\qquad
\frac{\vdash \Gamma, B \quad \vdash \Delta, \overline{B}}{\vdash \Gamma, \Delta}\ \textit{cut}
$$

Figure 2. Three rules admissible in $\bar{\bar{\mu}}$MALL

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

Figure 3. Two structural rules

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 would split into the two rules

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

That is, the 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 *coinvariant* of that rule.

We make the following observations about this proof system.

1. 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.

2. The proof rules for equality guarantee that function symbols are all treated injectively; thus, function symbols will act only as term constructors. In this paper, the only function symbols we

employ are for zero and successor: of course, a theory of arithmetic should treat these symbols injectively.

3. The admissibility of the three rules in Figure 2 for $\bar{\bar{\mu}}$MALL is proved in [3]. The general form of the initial rule is admissible, although the proof system only dictates a limited form of that rule via the $\mu\nu$ rule. 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 coinvariant $S = B(\nu B)$.

4. While the weakening and contraction rules are not generally admissible in $\bar{\bar{\mu}}$MALL, they are both admissible for $\mathbf{N}_1$-formulas, a fact that plays an important role in Section 7.

We could add the inference rules for equality, non-equality, and least and greatest fixed points to Gentzen's LK proof system for first-order classical logic [22]. We take a different approach, however, in that, we will only consider proof systems for classical logic using polarized versions of classical formulas. In particular, the $\bar{\bar{\mu}}\mathrm{LK}_p^+$ proof system is the result of adding to the $\bar{\bar{\mu}}$MALL proof system the rules for contraction C and weakening W from Figure 3 as well as the cut rule.

3.2. Examples

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

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

These polarized formulas belong to $\mathbf{N}_2$ and $\mathbf{N}_3$, respectively. Only the first of these is provable in $\bar{\bar{\mu}}$MALL, although both formulas are provable in $\bar{\bar{\mu}}\mathrm{LK}_p$.

Note that it is clear that if there exists a $\bar{\bar{\mu}}$MALL proof of a $\mathbf{P}_1$-formula, then that proof does not contain the ν rule, i.e., it does not contain the rules involving coinvariants. Finally, given that first-order Horn clauses can interpret Turing machines [44], and given that Horn clauses can easily be encoded using $\mathbf{P}_1$-formulas, it is undecidable whether or not a $\mathbf{P}_1$ expression has a $\bar{\bar{\mu}}$MALL proof. Similarly, $\mathbf{P}_1$ formulas can be used to specify any general recursive function. Obviously, the provability of $\mathbf{N}_1$-formulas is also undecidable.

The following are proofs of two axioms of Peano Arithmetic (see also Section 6).

$$\dfrac{\dfrac{}{\vdash (s\,x) \neq \mathbf{0}}\neq}{\vdash \forall x.\,(s\,x) \neq \mathbf{0}}\forall \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash x = x}=}{\vdash s\,x \neq s\,y, x = y}\neq}{\vdash (s\,x \neq s\,y) \parr (x = y)}\parr}{\vdash \forall x \forall y.\,(s\,x = s\,y) \supset (x = y)}\forall \times 2$$

The unary relation for denoting the set of natural numbers and the ternary relations for addition and multiplication can be axiomatized using Horn clauses as follows.

$$\mathit{nat}\ \mathbf{0}$$
$$\forall x(\mathit{nat}\ x \supset \mathit{nat}\ (s\ x))$$

$$\forall x(\mathit{plus}\ \mathbf{0}\ x\ x)$$
$$\forall x \forall y \forall u(\mathit{plus}\ x\ y\ u \supset \mathit{plus}\ (s\ x)\ y\ (s\ u))$$

$$\forall x(\mathit{mult}\ \mathbf{0}\ x\ \mathbf{0})$$
$$\forall x \forall y \forall u \forall w(\mathit{mult}\ x\ y\ u \wedge \mathit{plus}\ y\ u\ w \supset \mathit{mult}\ (s\ x)\ y\ w)$$

These Horn clauses can be mechanically transformed into the following least fixed point definitions of these relations.

$$\mathit{nat} = \mu\lambda N \lambda x(x = \mathbf{0} \oplus \exists x'(x = (s\ x') \otimes N\ x'))$$
$$\mathit{plus} = \mu\lambda P \lambda x \lambda y \lambda u((x = \mathbf{0} \otimes y = u) \oplus \exists x' \exists u' \exists w(x = (s\ x') \otimes u = (s\ u') \otimes P\ x'\ y\ u'))$$
$$\mathit{mult} = \mu\lambda M \lambda x \lambda y \lambda w\big((x = \mathbf{0} \otimes u = \mathbf{0}) \oplus \exists x' \exists u' \exists w(x = (s\ x') \otimes \mathit{plus}\ y\ u'\ w \otimes M\ x'\ y\ w)\big)$$

Both of these fixed point expressions are $\mathbf{P}_1$.

The following derivation verifies that 4 is a sum of 2 and 2.

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

To complete this proof, we must construct the (obvious) proof of $\vdash\ \mathit{nat}\ \mathbf{4}$ and a similar subproof verifying that $1 + 2 = 3$. Note that in the bottom-up construction of this proof, the witness used to instantiate the final $\exists p$ is, in fact, the sum of 2 and 2. Thus, this proof construction does not compute this sum's value but simply checks that 4 is the correct value.

In contrast to the above example, the following proof of $\forall u(\mathit{plus}\ \mathbf{2}\ \mathbf{2}\ u \supset \mathit{nat}\ u)$ can be seen as a *computation* of the value of 2 plus 2. The proof of this sequent begins as follows.

$$\dfrac{\dfrac{\dfrac{}{\vdash \mathbf{2} \neq \mathbf{0} \parr \mathbf{2} \neq u, \mathit{nat}\ u}\,\parr, \neq \quad \dfrac{\dfrac{\vdash \overline{\mathit{plus}\ \mathbf{1}\ \mathbf{2}\ u'}, \mathit{nat}\ (s\ u')}{\vdash \mathbf{2} \neq (s\ x'), u \neq (s\ u'), \overline{\mathit{plus}\ x'\ \mathbf{2}\ u'}, \mathit{nat}\ u}\,\neq \times 2}{\vdash \forall x' \forall u' \forall w(\mathbf{2} \neq (s\ x') \parr u \neq (s\ u') \parr \overline{\mathit{plus}\ x'\ \mathbf{2}\ u'}), \mathit{nat}\ u}\,\forall, \parr}{\vdash \overline{\mathit{plus}\ \mathbf{2}\ \mathbf{2}\ u}, \mathit{nat}\ u}\,\mathit{unfold}, \&}{\vdash \forall u(\overline{\mathit{plus}\ \mathbf{2}\ \mathbf{2}\ u} \parr \mathit{nat}\ u)}\,\forall, \parr$$

Similarly, the open premise above has a partial proof which reduces its provability to the provability of the sequent $\vdash\ \overline{\mathit{plus}\ \mathbf{0}\ \mathbf{2}\ u'}, \mathit{nat}\ (s\ (s\ u'))$. This final sequent is similarly reduced to $\vdash\ \mathbf{0} \neq \mathbf{0}, \mathbf{2} \neq$

u', $\mathit{nat}\,(s\;(s\;u'))$, which is itself reduced to $\vdash \mathit{nat}\,\mathbf{4}$, which has a trivial proof. Note that the bottom-up construction of this proof involves the systematic computation of the value of 2 plus 2.

The previous two proofs involved with the judgment $2 + 2 = 4$ illustrates two different ways to determine $2 + 2$: the first involves a "guess-and-check" approach, while the second involves a direct computation. We will return to these two approaches in Section 4.

Unpolarized formulas that state the *totality* and *determinacy* of the function encoded by a binary relation ϕ can be written as

$$[\forall x.\mathit{nat}\,x \supset \exists y.\mathit{nat}\,y \wedge \phi(x, y)] \wedge$$
$$[\forall x.\mathit{nat}\,x \supset \forall y_1.\mathit{nat}\,y_1 \supset \forall y_2.\mathit{nat}\,y_2 \supset \phi(x, y_1) \supset \phi(x, y_2) \supset y_1 = y_2].$$

If this formula is polarized so that the two implications are encoded using $⅋$, the conjunction is replaced by $\&$, and the expression ϕ is $\mathbf{P}_1$, then this formula is an $\mathbf{N}_2$ formula.

Given the definition of addition on natural numbers above, the following totality and determinacy formulas

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

can be proved in $\bar{\bar{\mu}}$MALL where $\supset$ is polarized using $⅋$ and the one occurrence of conjunction above is polarized using $\&$. These proofs require both induction and the $\mu\nu$ rule.

The direct connection between proof search in $\bar{\bar{\mu}}$MALL and the model checking problems of reachability and bisimilarity (and their negations) has been demonstrated in [27]. In particular, reachability problems were encoded as $\mathbf{P}_1$-formulas, while non-reachability problems were encoded as $\mathbf{N}_1$-formulas. The paper [27] also showed that the specification of simulation and bisimulation can be encoded as $\mathbf{N}_2$- formulas. Another common form of $\mathbf{P}_1$-formulas arises when applying the Clark completion [13] to Horn clause specifications.

3.3. Some known results concerning $\bar{\bar{\mu}}$MALL

While $\bar{\bar{\mu}}$MALL does not contain the contraction rule, it is still possible for the number of occurrences of logical connectives to grow in sequents when searching for a proof. In particular, the unfolding rule (when read from conclusion to premise) can make a sequent containing $(\mu B \vec{t})$ into a sequent containing $(B(\mu B)\vec{t})$: here, the abstracted formula B is repeated. Surprisingly, however, the subset of $\bar{\bar{\mu}}$MALL that does not contain occurrences of fixed points is still undecidable. In particular, consider the following two sets of inductively defined classes of $\bar{\bar{\mu}}$MALL formulas.

$$\Phi \;::=\; \Phi \,\&\, \Phi \mid \exists x.\Phi \mid \forall x.\Phi \mid \Psi$$
$$\Psi \;::=\; t_1 = t_1' \multimap \cdots \multimap t_n = t_n' \multimap t_0 = t_0' \quad (n \geq 0)$$

If we also assume that there are exactly three constructors, one each of type $\iota \to \iota$, $\iota \to \iota \to \iota$, and $\iota \to \iota \to \iota \to \iota$, then it is undecidable whether or not a given formula Φ is provable in $\bar{\bar{\mu}}$MALL [38].

The two main proof-theoretic results concerning $\bar{\bar{\mu}}$MALL are the admissibility of the cut rule (in Figure 2) and the completeness of a focusing proof system [3].

3.4. Definable exponentials

As Baelde showed in [3], the following definitions

$$?P = \mu(\lambda p.\bot \oplus (p \parr p) \oplus P) \qquad !P = \overline{?(\overline{P})}$$

approximate the exponentials of linear logic in the sense that the following four rules—dereliction, contraction, weakening, and promotion—are admissible in $\bar{\bar{\mu}}$MALL.

$$\frac{\vdash \Gamma}{\vdash ?B, \Gamma}\ W \qquad \frac{\vdash ?B, ?B, \Gamma}{\vdash ?B, \Gamma}\ C \qquad \frac{\vdash B, \Gamma}{\vdash ?B, \Gamma}\ D \qquad \frac{\vdash B, ?\Gamma}{\vdash !B, ?\Gamma}\ P$$

In particular, we use $\bar{\bar{\mu}}$MALL$(!, ?)$ to denote the extension of $\bar{\bar{\mu}}$MALL with the two exponentials ! and ? and the above four proof rules. Thus, every $\bar{\bar{\mu}}$MALL$(!, ?)$-provable sequent can be mapped to a $\bar{\bar{\mu}}$MALL-provable sequent by simply replacing the exponentials for their corresponding fixed point definition.

4. Using proof search to compute functions

We say that a binary relation ϕ encodes a function f if $\phi(x, y)$ holds if and only if $f(x) = y$. Of course, this correspondence is only well-defined if we know that the *totality* and *determinacy* properties hold for ϕ. For example, let *plus* be the definition of addition on natural numbers given in Section 3. The following polarized formulas encoding totality and determinacy are $\mathbf{N}_2$-formulas.

$$[\forall x_1 \forall x_2.\ \mathit{nat}\ x_1 \multimap \mathit{nat}\ x_2 \multimap \exists y. \mathit{nat}\ y \otimes \mathit{plus}\ x_1\ x_2\ y]$$
$$[\forall x_1 \forall x_2.\ \mathit{nat}\ x_1 \multimap \mathit{nat}\ x_2 \multimap \forall y_1 \forall y_2.\ \mathit{plus}\ x_1\ x_2\ y_1 \multimap \mathit{plus}\ x_1\ x_2\ y_2 \multimap y_1 = y_2]$$

These formulas can be proved in $\bar{\bar{\mu}}$MALL.

One approach to computing the function that adds two natural numbers is to follow the Curry-Howard approach of relating proof theory to computation [30]. First, extract from a natural deduction proof of the totality formula above a typed λ-term. Second, apply that λ-term to the λ-terms representing the two proofs of, say, *nat* n and *nat* m. Third, use a nondeterministic 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. The resulting normal λ-term should encode the proof of *nat* p, where p is the sum of n and m. In this section, we will present an alternative mechanism for computing functions from their relational specification that relies on using proof search mechanisms instead of this proof-normalization mechanism.

The totality and determinacy properties of some binary relation ϕ can be expressed equivalently as, for any natural number n, the expression $\lambda y.\phi(n, y)$ denotes a singleton set. Of course, the sole member of that singleton set is the value of the function it encodes. If our logic contained a choice operator, such as Church's *definite description* operator ι [12], this function can be represented as $\lambda x.\iota y.\phi(x, y)$. The search for proofs can, however, be used to provide a more computational approach to computing the function encoded by ϕ. Assume that P and Q are predicates of arity one and that

P denotes a singleton. In this case, the (unpolarized) formulas $\exists x[Px \wedge Qx]$ and $\forall x[Px \supset Qx]$ are logically equivalent, although the proof search semantics of these 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 $\forall x[Px \supset Qx]$, we allocate an eigenvariable y and attempt to prove the sequent $\vdash Py \supset Qy$. Such an attempt at building a proof might *compute* the value t (especially if we can restrict proofs of that implication not to involve the general form of induction). This difference was illustrated in Section 3 with the proof of $\vdash \exists p.\mathit{plus}\ \mathbf{2}\ \mathbf{2}\ p \otimes \mathit{nat}\ p$ (which guesses and checks that the value of 2 plus 2 is 4) versus the proof of $\forall u(\overline{\mathit{plus}\ \mathbf{2}\ \mathbf{2}\ u} \mathbin{⅋} \mathit{nat}\ u)$ (which incrementally constructs the sum of 2 and 2).

Assume that P is a $\mathbf{P}_1$ predicate expression of type $i \rightarrow o$ and that we have a $\bar{\bar{\mu}}$MALL proof of $\forall x[Px \supset \mathit{nat}\ x]$. If this proof does not contain the induction rule, then that proof can be seen as computing the sole member of P. As the following example shows, it is not the case that if there is a $\bar{\bar{\mu}}$MALL proof of $\forall x[Px \supset \mathit{nat}\ x]$ then it has a proof in which the only form of the induction rule is unfolding. To illustrate this point, 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 $\bar{\bar{\mu}}$MALL proof that $\forall x[Px \supset \mathit{nat}\ x]$, 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 follows the outline

$$\dfrac{\vdash \mathit{nat}\ \mathbf{0} \qquad \dfrac{\dfrac{\vdots}{\vdash \overline{P\ (s\ (s\ y))}, \mathit{nat}\ y}}{\vdash \overline{P\ (s\ y)}, \mathit{nat}\ y}\ \mathit{unfold}, \&, \neq}{\vdash \overline{P\ y}, \mathit{nat}\ y}\ \mathit{unfold}, \&, \neq .$$

Although proof search can contain potentially unbounded branches, we can still use the proof search concepts of unification and nondeterministic search to compute the value within a singleton. We now define a nondeterministic algorithm to do exactly that. 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 $\mathbf{P}_1$-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 nondeterministically 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 nondeterministic algorithm is essentially applying left-introduction rules (assuming a two-sided sequent calculus) in a bottom-up fashion and, if there are two premises, selecting (nondeterministically) just one premise to follow.

Lemma 4.1. Assume that P is a $\mathbf{P}_1$ expression of type $i \to o$ and that $\exists y.Py$ has a $\bar{\mu}$MALL 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 (cut-free) $\bar{\mu}$MALL proof of $\theta(B_i)$.

Note that we are left with a regular state if we strike out the augmented items. Given that we have a $\bar{\mu}$MALL proof of $\exists y.Py$, we must have a $\bar{\mu}$MALL proof Ξ_0 of $P\ t$ for some term t. Note that there is no occurrence of the induction rule 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 nondeterministic 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 $\theta = \varphi \circ \rho$.

We must show that the transition is made to an augmented state in each of these cases. 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. $\square$

This lemma ensures that our search algorithm can compute a member from a non-empty set, given a $\bar{\bar{\mu}}$MALL 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 4.2. Assume that P is a $\mathbf{P}_1$ expression of type $i \to o$ and that $\exists! y.Py$ has a $\bar{\bar{\mu}}$MALL 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 $\bar{\bar{\mu}}$MALL proof.

Proof:
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 $\bar{\bar{\mu}}$MALL proof of $P\ t$ for some term t and, hence, of $\exists y.P\ y$. By Lemma 4.1, there is a sequence of transitions from the 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 a (cut-free) $\bar{\bar{\mu}}$MALL proof of $\exists! y.Py$, that proof contains a $\bar{\bar{\mu}}$MALL proof of $\forall x_1 \forall x_2.P\ x_1 \supset P\ x_2 \supset x_1 = x_2$, which, when combined using cut with the proofs of the formulas Pt and Ps (and the admissibility of cut for $\bar{\bar{\mu}}$MALL) allows us to conclude that $t = s$. $\square$

Thus, a (naive) proof-search algorithm involving both unification and nondeterministic search is sufficient for computing the functions encoded in relations in this setting. This result puts the computation of such functions inside the domain of logic programming, where relations, unification, and nondeterministic proof search are routinely encountered. As a result, deploying any number of Prolog-style implementation strategies, such as those found in [1, 43], can make the search for such proofs more effective.

5. The totality of the Ackermann function

The question of the expressivity of $\bar{\bar{\mu}}$MALL has been analyzed by Baelde [2, Section 3.5], who provided a lower bound to it by characterizing a subset of $\bar{\bar{\mu}}$MALL where proofs can be interpreted as primitive recursive functions, and cut elimination corresponds to computing those functions. The ideas behind the encoding can be used in order to express primitive recursive functions as fixed points and provide proofs of the totality of these functions in a similar fashion to that used for the $plus$ relation.

However, Baelde noted that this encoding is insufficient for a computational interpretation of Ackermann's function. Furthermore, it is also insufficient to obtain a proof that the underlying relation represents a total function.

We show here a different method, based on the extension to $\bar{\bar{\mu}}$MALL$(!,?)$ provided in Section 3.4, that allows us to prove the totality of Ackermann's function. The encoding of Ackermann's function in $\bar{\bar{\mu}}$MALL is based on the following relational specification.

$$\begin{aligned} ack = \mu(\lambda ack \lambda m \lambda n \lambda a.\ & (m = 0 \otimes a = s\ n) \oplus \\ & \exists p (m = s\ p \otimes n = 0 \otimes ack\ p\ (s\ 0)\ a) \oplus \\ & \exists p \exists q \exists b (m = s\ p \otimes n = s\ q \otimes ack\ m\ q\ b \otimes ack\ p\ b\ a)) \end{aligned}$$

In order to prove that this three-place relation determines a total function, we need to prove *determinacy* (the first two arguments uniquely determine the third argument) and *totality* (for every choice of the first two arguments, there exists a value for the third argument). The proof of determinacy, that is, of the formula

$$\forall x \forall y \forall a_1 \forall a_2.(ack\ x\ y\ a_1 \multimap ack\ x\ y\ a_2 \multimap a_1 = a_2),$$

proceeds simply as follows: first, perform induction on $ack\ x\ y\ a_1$ using the rest of the context as an invariant; second, perform case analysis on the three ways that $ack\ x\ y\ a_1$ is defined, and, third, in each case use the inductive assumption to complete the proof. We have described an $\bar{\bar{\mu}}$MALL proof since neither contraction nor weakening is needed in this proof. (This proof outline also applies to proving the determinacy of the *plus* relation in Section 3.2.) In the rest of this section, we prove the following formula regarding the totality of this relation.

$$\forall m \forall n (nat\ m \multimap nat\ n \multimap \exists a.\ (ack\ m\ n\ a \otimes nat\ a))$$

We now illustrate a proof of this formula.[1] In doing so, we will highlight the crucial use of the encoded exponentials in $\bar{\bar{\mu}}$MALL$(!,?)$. For greater clarity, we will use $\multimap$ as a shorthand, and we will retain the overline syntax for negation instead of computing the explicit De Morgan duality. The proof begins by introducing the universal quantifiers and then applying twice the $\parr$ rule:

$$\dfrac{\vdash \overline{nat\ m}, \overline{nat\ n}, \exists a.\ ack\ m\ n\ a}{\forall m \forall n (nat\ m \multimap nat\ n \multimap \exists a.\ ack\ m\ n\ a)}\ \forall, \parr$$

At this point, we need to use the coinduction rule twice, once with $\overline{nat\ n}$ and once with $\overline{nat\ m}$. The coinvariants we introduce for these inductions will be where we exploit the encoding of the exponentials. In the first induction, we use as coinvariant the negation of the remaining context of the sequent with a ! added, that is $\overline{\lambda m !(\forall n\ nat\ n \multimap \exists a(ack\ m\ n\ a \otimes nat\ a))}$. This coinvariant needs to be contracted later in the proof, hence the need for the exponential. The left premise of the ν rule is immediately verified since the coinvariant starts with a ? which we can derelict away, and we can conclude immediately after by using the fact that a generalized initial rule $\vdash \Gamma, \Gamma^{\perp}$ is admissible in

[1] A formalization of this proof using the Abella theorem prover [4] is available at
`https://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/PA-and-muMALL/`.

$\bar{\bar{\mu}}$MALL.[2] The right premise of the ν rule then yields the base and inductive steps. In the base case, we need to prove $\vdash\ !(\forall n\ nat\ n \multimap \exists a(ack\ \mathbf{0}\ n\ a \otimes nat\ a))$, and we do this by promoting away the exponential and then unfolding the base case of the ack definition. The inductive step gives us:

$$\dfrac{\vdash \overline{!(\forall n\ nat\ n \multimap \exists a(ack\ x\ n\ a \otimes nat\ a))}, \overline{nat\ n}, \exists a(ack\ (s\ x)\ n\ a \otimes nat\ a)}{\vdash \overline{!(\forall n\ nat\ n \multimap \exists a(ack\ x\ n\ a \otimes nat\ a))}, !(\forall n\ nat\ n \multimap \exists a(ack\ (s\ x)\ n\ a \otimes nat\ a))}\ !, \forall\, \invamp$$

Since the coinvariant starts with a question mark, we can promote the dualized coinvariant and continue the proof. We have again exposed the dualized $\overline{nat\ n}$ predicate, over which we can perform the second induction. As before, we take the entire sequent (abstracted over by n) and negate it, but this time there is no need to add another occurrence of an exponential, obtaining the coinvariant

$$\overline{\lambda k !(\forall n\ nat\ n \multimap \exists a(ack\ x\ n\ a \otimes nat\ a)) \multimap \exists a(ack\ (s\ x)\ k\ a \otimes nat\ a)}.$$

The left-hand premise of the ν rule is now exactly an instance of $\vdash \Gamma, \Gamma^{\perp}$. The base case and the inductive steps for this second induction remain to be proved. The base case (where we need to prove the coinvariant for k being 0) is again proved by a routine inspection of the definition of ack. The antecedent part of the coinvariant can be used directly since it starts with $?$.

The final step is the coinductive case, where we need to prove the invariant for $(s\ k)$ given the invariant for k: that is, we need to prove the sequent

$$\begin{array}{l}\vdash \overline{!\,\forall x(nat\ x \multimap \exists a(ack\ y\ x\ a \otimes nat\ a)) \multimap \exists a(ack\ (s\ y)\ k\ a \otimes nat\ a)},\\ \quad !\,\forall x(nat\ x \multimap \exists a(ack\ y\ x\ a \otimes nat\ a)) \multimap \exists a(ack\ (s\ y)\ (s\ k)\ a \otimes nat\ a)\end{array}$$

Introducing the second linear implication gives the dual of the antecedent, which starts with $?$ and, hence, is a contractable copy of the coinvariant from the previous induction:

$$?\,\overline{\forall x(nat\ x \multimap \exists a(ack\ y\ x\ a \otimes nat\ a))}.$$

The entire reason for using $\bar{\bar{\mu}}$MALL$(!, ?)$ to state coinvariants in this proof is to make this contraction possible. Now, we can decompose the new coinvariant, a universally quantified implication, and use the two copies we have obtained: one copy is provided to the antecedent of the implication, and one copy is used to continue the proof. The two premises of this occurrence of the ν rule are:

$$\vdash !\forall x(nat\ x \multimap \exists a(ack\ y\ x\ a \otimes nat\ a)), ?\overline{\forall x(nat\ x \multimap \exists a(ack\ y\ x\ a \otimes nat\ a))}$$

$$\vdash \overline{\exists a(ack\ (s\ y)\ k\ a \otimes nat\ a)}, ?\overline{\forall x(nat\ x \multimap \exists a(ack\ y\ x\ a \otimes nat\ a))}, \exists a(ack\ (s\ y)\ (s\ k)\ a \otimes nat\ a)$$

The first one is immediately proved thanks to the fact that the exponentials are dual. The second sequent is also easily proved by unfolding the definition of ack using its third case; the exponential can be derelicted, and all the arising premises can be proved without the exponentials.

Given that we have a $\bar{\bar{\mu}}$MALL$(!, ?)$ proof (hence, also a $\bar{\bar{\mu}}$MALL proof) of the totality of the Ackermann relation, we can use the proof search method in Section 4 in order to actually compute the Ackermann function. Additionally, from the cut-elimination theorem of $\bar{\bar{\mu}}$MALL, we obtain an interpretation as a computation via proof normalization.

[2] If Γ is the multiset $\{B_1, \ldots, B_n\}$ then $\Gamma^{\perp}$ is $\{\overline{B_1} \otimes \cdots \otimes \overline{B_n}\}$.

6. $\bar{\mu}\mathrm{LK}_p^+$ contains Peano Arithmetic

We now turn our attention to $\bar{\mu}\mathrm{LK}_p^+$: recall from Section 3.1 that this proof system is the result of adding to the $\bar{\mu}$MALL proof system both contraction C and weakening W (see Figure 3) as well as the cut rule. The consistency of $\bar{\mu}$MALL follows immediately from its cut-elimination theorem. It is worth noting that adding contraction to some consistent proof systems with weak forms of fixed points can make the new proof system inconsistent. For example, both Girard [25] and Schroeder-Heister [41] 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. (See also [26].) The following theorem proves that adding contraction to $\bar{\mu}$MALL does not lead to inconsistency.

Theorem 6.1. $\bar{\mu}\mathrm{LK}_p^+$ is consistent: that is, the empty sequent is not provable.

Proof:
Consider the sequent $\vdash B_1, \ldots, B_n$, where $n \geq 0$ and where all the free variables of formulas in this sequent are contained in the list of variables $x_1, \ldots, x_m$ ($m \geq 0$) all of type ι. We say that this sequent is *true* if for all substitutions θ that send the variables $x_1, \ldots, x_m$ to closed terms of type ι (numerals), the disjunction of the unpolarized versions of the formula $B_1\theta, \ldots B_n\theta$ is *true* (in the standard model). (The empty disjunction is the same as false.) A straightforward induction on the structure of $\bar{\mu}\mathrm{LK}_p^+$ proofs shows that all of the inference rules in Figures 1, 2, and 3 are sound (meaning that when the premises are true, the conclusion is true). Thus, we have the following soundness result: if the sequent $\vdash B_1, \ldots, B_n$ is provable in $\bar{\mu}\mathrm{LK}_p^+$, then that sequent is true. As a result, the empty sequent is not provable. □

We now show that Peano arithmetic is contained in $\bar{\mu}\mathrm{LK}_p^+$. The terms of Peano arithmetic are identical to the terms introduced in Section 2 for encoding numerals. The formulas of Peano arithmetic are similar to unpolarized formulas except that they are built from $=$, $\neq$, the propositional logical connectives $\wedge$, *tt*, $\vee$, *ff*, and the two quantifiers $\hat{\forall}$ and $\hat{\exists}$ (both of type $(i \to o) \to o$). Such formulas can be *polarized* to get a polarized formula as described in Section 2.3. Finally, all occurrences of $\hat{\forall}$ and $\hat{\exists}$ are replaced by $\lambda B.\forall x\ (\overline{\mathit{nat}\ x} \parr (Bx))$ and $\lambda B.\exists x\ (\mathit{nat}\ x \otimes (Bx))$, respectively. Here, *nat* is an abbreviation for $\mu\lambda N\lambda n(n = \mathbf{0} \oplus \exists m(n = (s\ m) \otimes N\ m))$.

Most presentations of Peano arithmetic incorporate the addition and multiplication of natural numbers as binary function symbols or as three-place relations. We will avoid introducing the extra constructors $+$ and $\cdot$ and choose to encode addition and multiplication as relations. In particular, these are defined as the fixed point expressions *plus* and *mult* given in Section 3. The relation between these two presentations is such that the equality $x + y = w$ corresponds to *plus* $x\ y\ w$ and the equality $x \cdot y = w$ corresponds to *mult* $x\ y\ w$. A more complex expression, such as $\forall x \forall y.\ (x \cdot s\ y = (x \cdot y + x))$, can similarly be written as either

$$\forall x \forall y \forall u.\ \mathit{mult}\ x\ (s\ y)\ u \supset\ \forall v.\ \mathit{mult}\ x\ y\ v \supset \forall w.\mathit{plus}\ v\ x\ w \supset u = w$$

or as

$$\forall x \forall y \exists u.\ \mathit{mult}\ x\ (s\ y)\ u \wedge\ \exists v.\ \mathit{mult}\ x\ y\ v \wedge \exists w.\mathit{plus}\ v\ x\ w \wedge u = w.$$

A general approach to making such an adjustment to the syntax of expressions using functions symbols to expressions using relations is discussed from a proof-theoretic perspective in [23].

Proofs in Peano arithmetic can be specified using the following six axioms.

$$\begin{array}{ll} \forall x.\, (s\, x) \neq \mathbf{0} & \forall x \forall y.\, (x + s\, x) = s(x+y) \\ \forall x \forall y.\, (s\, x = s\, y) \supset (x = y) & \forall x.\, (x \cdot \mathbf{0} = \mathbf{0}) \\ \forall x.\, (x + \mathbf{0} = x) & \forall x \forall y.\, (x \cdot s\, y = (x \cdot y + x)) \end{array}$$

and the axiom scheme (which we write using the predicate variable A)

$$(A\mathbf{0} \wedge \forall x.\, (Ax \supset A(s\, x))) \supset \forall x.\, Ax.$$

We also admit the usual inference rules of modus ponens and universal generalization.

Theorem 6.2. ($\bar{\bar{\mu}}\mathrm{LK}_p^+$ 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 $\hat{Q}$ is provable in $\bar{\bar{\mu}}\mathrm{LK}_p^+$.

Proof:
It is easy to prove that *mult* and $plus$ describe precisely the multiplication and addition operations on natural numbers. As we illustrate next, the translations of the Peano Axioms can all be proved in $\bar{\bar{\mu}}\mathrm{LK}_p^+$. Since the presence of contraction and weakening in $\bar{\bar{\mu}}\mathrm{LK}_p^+$ means that different polarizations of a formula are all equivalence in $\bar{\bar{\mu}}\mathrm{LK}_p^+$, we only need to consider proving a single such polarization. The following formulas result from polarizing the translation of the first two Peano Axioms.

$$\forall x. \overline{\mathit{nat}\, x} \parr (s\, x) \neq \mathbf{0} \quad \text{and} \quad \forall x. \overline{\mathit{nat}\, x} \parr \forall y. \overline{\mathit{nat}\, y} \parr \overline{(s\, x = s\, y)} \parr (x = y).$$

Given that induction is not needed to prove these formulas and that the weakening rule is admissible for $\mathbf{N}_1$-formulas (as observed in Section 3.1 and proved in [3, 7]), the proof of these two formulas are essentially the same as the proofs given for their untranslated forms in Section 3.2. The proofs of the next four axioms use induction in the usual way. Thus, consider the final axiom—the induction scheme—and it polarized translation

$$\overline{(A\mathbf{0} \otimes \forall x.\, (\overline{nat\ x} \parr \overline{Ax} \parr A(s\, x)))} \parr \forall x.\, (\overline{nat\ x} \parr Ax)$$

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 $\bar{\bar{\mu}}\mathrm{LK}_p^+$ allows us to encode modus ponens. □

7. Conservativity results for linearized 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 [20].

We present two conservativity theorems in this section that relate the stronger logic $\bar{\bar{\mu}}$LK$_p$ to the weaker logic $\bar{\bar{\mu}}$MALL.

The strongest set of theorems we explore in this section are based on $\mathbf{N}_2$-formulas, and the richest sequents we consider contain only $\mathbf{N}_1$, $\mathbf{P}_1$, and $\mathbf{N}_2$-formulas: such sequents are called *reduced*. Note that the sequent $\vdash B_1, \ldots, B_n$ $(n \geq 2)$ is reduced if and only if the formula $B_1 \parr \cdots \parr B_n$ is $\mathbf{N}_2$.

An $\mathbf{N}_2$-formula is *pointed* if every occurrence of $B_1 \parr B_2$ in it is such that either B_1 or B_2 is $\mathbf{N}_1$ and every occurrence of νB in it is such that B is $\mathbf{N}_1$. This notion of pointed has been used in game-theoretic semantics for linear logic [18] (in the form of *simple expressions*) and model checking [27] (where it was called *switchable*). In the context of $\bar{\bar{\mu}}$MALL, focusing on a pointed formula results in additive synthetic rules even when that formula contains multiplicative connectives [27]. The formulas in $\mathbf{N}_1$ and $\mathbf{P}_1$ are pointed. Also, if B_0 is in $\mathbf{P}_1$ then the formula $\forall \vec{x}.B_1 \multimap \cdots \multimap B_n \multimap B_0$ $(n \geq 1)$ is a pointed formula if and only if the formulas $B_1, \ldots, B_n$ are all in $\mathbf{P}_1$. The formulas stating the totality and determinacy of the *plus* relation in Section 4 and the totality of Ackermann's relation in Section 5 are pointed formulas. We say that a reduced sequent $\vdash B_1, \ldots, B_n$ is *pointed* if $n = 1$ and B_1 is pointed or $n \geq 2$ and $B_1 \parr \cdots \parr B_n$ is pointed. Put in an equivalent way: the reduced sequent $\vdash B_1, \ldots, B_n$ is pointed if and only if it contains at most one pointed formula that is not $\mathbf{N}_1$.

7.1. $\bar{\bar{\mu}}$LK$_p$ is conservative over $\bar{\bar{\mu}}$MALL for $\mathbf{P}_1$-formulas

A *positive region* is a (cut-free) $\bar{\bar{\mu}}$LK$_p$ proof that contains only the inference rules $\mu\nu$, contraction, weakening, and the introduction rules for the positive connectives: *i.e.*, there are no introduction rules for the negative connectives. If $\vdash \Gamma_1$ and $\vdash \Gamma_2$ are sequents, we say that $\vdash \Gamma_1$ is a *subsequent of* $\vdash \Gamma_2$ if Γ_1 is a sub-multiset of Γ_2.

Lemma 7.1. Let $\vdash \Gamma$ be a reduced sequent that has a positive region proof. There exists a pointed subsequent $\vdash \Gamma'$ of $\vdash \Gamma$ such that $\vdash \Gamma'$ has a $\bar{\bar{\mu}}$MALL proof.

Proof:
Let Ξ be a positive region proof of $\vdash \Gamma$. We proceed by induction on the structure of Ξ.

If Ξ is the $\mu\nu$ rule, then Γ contains exactly two occurrences of formulas that are complementary: since this sequent is reduced, one of these formulas is positive and, hence, must be $\mathbf{P}_1$ and the other formula is $\mathbf{N}_1$. Therefore, $\vdash \Gamma$ is pointed and has Ξ as a $\bar{\bar{\mu}}$MALL proof.

Next, consider the case where the last inference rule of Ξ is either the following contraction or weakening (where Γ is $\{B\} \cup \Gamma_0$).

$$\frac{\vdash \Gamma_0}{\vdash \Gamma_0, B}\ W \qquad \frac{\vdash \Gamma_0, B, B}{\vdash \Gamma_0, B}\ C$$

If B is positive, the inductive assumption yields the conclusion immediately. If B is an $\mathbf{N}_1$-formula, then the result follows from the inductive assumption and the admissibility in $\bar{\bar{\mu}}$MALL of these structural rules for $\mathbf{N}_1$-formulas.

Finally, consider the case where the last rule of Ξ is an introduction rule for one of the positive connectives $\otimes$, 1, $\oplus$, $=$, $\exists$, or μ. The most involved of these cases is when the last inference rule of Ξ

introduces $\otimes$. Thus, Γ can be written as $\{B_1 \otimes B_2\} \cup \Gamma_0$ (note that B_1 and B_2 are $\mathbf{P}_1$-formulas) and Ξ is of the form

$$\dfrac{\begin{array}{c}\Xi_1 \\ \vdash \Gamma_1, B_1\end{array} \quad \begin{array}{c}\Xi_2 \\ \vdash \Gamma_2, B_2\end{array}}{\vdash \Gamma_0, B_1 \otimes B_2} \otimes,$$

where Γ_0 equals $\Gamma_1 \cup \Gamma_2$. By the inductive hypothesis, there are pointed subsequents $\vdash \Delta_1$ and $\vdash \Delta_2$ of the left and right premises, respectively, such that $\vdash \Delta_1$ and $\vdash \Delta_2$ have $\bar{\bar{\mu}}$MALL proofs Ξ_1' and Ξ_2', respectively. If Δ_1 is a subsequent of Γ_1, we can take Γ' to be Δ_1. If Δ_2 is a subsequent of Γ_2, we can take Γ' to be Δ_2. The only remaining case is when Δ_1 is $\{B_1\} \cup \Delta_1'$ and Δ_2 is $\{B_2\} \cup \Delta_2'$ for multisets Δ_1' and Δ_2'. Note that Δ_1' and Δ_2' contain only $\mathbf{N}_1$-formulas. The sequent $\vdash \Delta_1, \Delta_2, B_1 \otimes B_2$ is then a pointed subsequent of $\vdash \Gamma$ with the $\bar{\bar{\mu}}$MALL proof that results from applying the $\otimes$ rule to Ξ_1' and Ξ_2'. ☐

Theorem 7.2. $\bar{\bar{\mu}}$LK$_p$ is conservative over $\bar{\bar{\mu}}$MALL for $\mathbf{P}_1$-formulas. That is, if B is a $\mathbf{P}_1$-formula and $\vdash B$ has a $\bar{\bar{\mu}}$LK$_p$ proof then $\vdash B$ has a $\bar{\bar{\mu}}$MALL proof.

Proof:
Let B be a $\mathbf{P}_1$-formula. If the sequent $\vdash B$ has a $\bar{\bar{\mu}}$LK$_p$ proof, that proof must be a positive region since it contains no negative subformulas. Thus, by Lemma 7.1, we know that that sequent contains a pointed sequent that is provable in $\bar{\bar{\mu}}$MALL proof. However, the only such sequent is $\vdash B$. ☐

7.2. $\bar{\bar{\mu}}\mathbf{LK}_p(\mathbf{N}_1)$ is conservative over $\bar{\bar{\mu}}$MALL for $\mathbf{N}_1$-formulas

Our next conservativity result requires restricting the complexity of coinvariants used in the ν rule. We say that a sequent has a $\bar{\bar{\mu}}$LK$_p(\mathbf{N}_1)$ proof if it has a $\bar{\bar{\mu}}$LK$_p$ proof in which all coinvariants of the proof are $\mathbf{N}_1$. This restriction on proofs is similar to the restriction that yields the $I\Sigma_1$ fragment of Peano Arithmetic [40]. Note that the proof of the totality of the Ackermann function discussed in Section 5 is an example of a proof that is not in $\bar{\bar{\mu}}$LK$_p(\mathbf{N}_1)$ since it uses complex coinvariants (involving the encoding of an exponential) that are much richer than $\mathbf{N}_1$.

Lemma 7.3. If the conclusion of a $\bar{\bar{\mu}}$LK$_p(\mathbf{N}_1)$ inference rule that introduces a negative connective is a pointed sequent then all of the premises of that rule are pointed.

Proof:
We illustrate this proof for three of the rules introducing negative connectives. The remaining cases are similar and simpler.

Assume that in the following inference rule, the sequent $\vdash \Gamma, B \parr C$ is pointed.

$$\dfrac{\vdash \Gamma, B, C}{\vdash \Gamma, B \parr C} \parr$$

We have three cases to consider. (i) All members of $\Gamma \cup \{B \parr C\}$ are $\mathbf{N}_1$-formulas. In that case, the premise contains only $\mathbf{N}_1$-formulas and is, thus, pointed. (ii) $B \parr C$ is not $\mathbf{N}_1$. Thus, it is $\mathbf{N}_2$. Since it is a pointed formula, either B is $\mathbf{N}_1$ and C is $\mathbf{N}_2$ or C is $\mathbf{N}_1$ and B is $\mathbf{N}_2$. In either case, the premise is pointed. (iii) There is a formula in Γ that is not $\mathbf{N}_1$. Thus, $B \parr C$ is $\mathbf{N}_1$ and so are B and C. The premise is again pointed.

Assume that in the following inference rule, the sequent $\vdash \Gamma, B \mathbin{\&} C$ is pointed.

$$\frac{\vdash \Gamma, B \quad \vdash \Gamma, C}{\vdash \Gamma, B \mathbin{\&} C}\ \&$$

We again have three cases to consider. (i) All members of $\Gamma \cup \{B \mathbin{\&} C\}$ are $\mathbf{N}_1$-formulas. In that case, the premises contain only $\mathbf{N}_1$-formulas and are, thus, pointed. (ii) $B \parr C$ is not $\mathbf{N}_1$. Thus, it is $\mathbf{N}_2$. Since it is a pointed formula, either B is $\mathbf{N}_1$ and C is $\mathbf{N}_2$ or C is $\mathbf{N}_1$ and B is $\mathbf{N}_2$. In either case, the premises are pointed. (iii) There is a formula in Γ that is not $\mathbf{N}_1$. Thus, $B \mathbin{\&} C$ is $\mathbf{N}_1$ and so are B and C. The premises are again pointed.

Assume that in the following inference rule, the sequent $\vdash \Gamma, \nu B\vec{t}$ is pointed.

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

Since we are in $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$, both B and S are in $\mathbf{N}_1$. Hence, the right premise is pointed. The left premise is also pointed since both $S\vec{t}$ and $\nu B\vec{t}$ are $\mathbf{N}_1$-formulas. □

Theorem 7.4. $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$ is conservative over $\bar{\bar{\mu}}$MALL for $\mathbf{N}_1$-formulas. That is, if B is an $\mathbf{N}_1$-formula and $\vdash B$ has a $\bar{\bar{\mu}}\mathrm{LK}_p$ proof, then $\vdash B$ has a $\bar{\bar{\mu}}$MALL proof.

Proof:
Let Γ be a multiset of $\mathbf{N}_1$-formulas and let Ξ be an $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$ proof of Γ. Note that all occurrences of formulas in all sequents in Ξ are in $\mathbf{N}_1$. We proceed by induction on the number of structural rules (weakening and contraction) that occur in Ξ. If this number is zero, then Ξ is the desired $\bar{\bar{\mu}}$MALL proof. Otherwise, assume that there is a structural rule and choose one uppermost occurrence. For example, if this structural rule is the contraction

$$\frac{\vdash N, N, \Delta}{\vdash N, \Delta}\ ,$$

then the premise has a $\bar{\bar{\mu}}$MALL proof. By the admissibility of contraction in $\bar{\bar{\mu}}$MALL for $\mathbf{N}_1$-formulas [3, Proposition 2.12], we know that $\vdash N, \Delta$ has a $\bar{\bar{\mu}}$MALL proof. A similar argument holds if the uppermost structural rule is weakening since weakening is similarly admissible for $\mathbf{N}_1$ formulas in $\bar{\bar{\mu}}$MALL. As a result, we can build a new $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$ proof Ξ' where this uppermost structural rule is replaced with a $\bar{\bar{\mu}}$MALL proof, thus reducing the number of structural rules from those in Ξ. □

7.3. $\bar{\bar{\mu}}\mathbf{LK}_p(\mathrm{N}_1)$ is conservative over $\bar{\bar{\mu}}$MALL for pointed formulas

We can conclude from the two preceding theorems that $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$ is conservative over $\bar{\bar{\mu}}$MALL for both $\mathbf{P}_1$ and $\mathbf{N}_1$-formulas. We can generalize these two conservativity results to $\mathbf{N}_2$-formulas if we restrict such formulas to be pointed.

Theorem 7.5. $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$ is conservative over $\bar{\bar{\mu}}$MALL for pointed $\mathbf{N}_2$-formulas. That is, if B is a pointed formula such that $\vdash B$ has a $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$ proof, then $\vdash B$ has a $\bar{\bar{\mu}}$MALL proof.

The proof of this theorem would be greatly aided if we had a focusing theorem for $\bar{\bar{\mu}}\mathrm{LK}_p$. If we take the focused proof system for $\bar{\bar{\mu}}$MALL given in [3, 7] and add contraction (in the decide rule) and weakening (in the initial rule), we have a natural candidate for a focused proof system for $\bar{\bar{\mu}}\mathrm{LK}_p$. (The focused proof systems LKF and MALLF in [33] have exactly these differences.) However, the completeness of that focused proof system is currently open. As Girard points out in [24], 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 $\bar{\bar{\mu}}\mathrm{LK}_p$, we must now reproduce aspects of focusing to prove Theorem 7.5. We shall return to prove this theorem at the end of this section.

Lemma 7.6. Let $\vdash \Gamma$ be a sequent containing only pointed formulas. Every formula occurrence in a sequent occurring in a $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$ proof of $\vdash \Gamma$ is pointed.

Proof:
Let $\vdash \Gamma$ be a sequent in which every formula is pointed, and let Ξ be a $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$ proof of this sequent. We now prove by induction on the structure of Ξ that every formula occurrence in every sequent in that proof is pointed. Since subformulas of a pointed formula are pointed, the only inference rules we need to check for this property explicitly are the rules for fixed points since they do not obey the subformula property. In particular, consider the derivation

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

In the ν rule, both B and S are $\mathbf{N}_1$ and, hence, $BS\vec{x}$ is $\mathbf{N}_1$ and $\overline{(S\vec{x})}$ is $\mathbf{P}_1$. As a result, the right premise of the ν rule contains only pointed formulas. The inductive assumption yields the same conclusion for the left premise. In the μ rule, the occurrence of $\mu B\vec{t}$ is $\mathbf{P}_1$ hence so is $B(\mu B)\vec{t}$. This case follows from the inductive hypothesis. □

Let $\vdash P, N, \Gamma$ be a sequent where P is a positive formula, and N is a negative formula. Assume that we have a proof of this sequent where the P formula is introduced at the root and the N formula is introduced on one of the premises of that rule. As is known from $\bar{\bar{\mu}}$MALL (see [3, 7]), all occurrences of a positive connective introduced immediately below the introduction of a negative connective can be permuted so that the negative connectives are introduced immediately below the positive connective. As a result, we shall introduce the following normal forms of proofs: an $\bar{\bar{\mu}}\mathrm{LK}_p$ proof is a P/N-proof if it is a $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$ proof in which there is no occurrence of a negative introduction rule above a positive introduction rule. By permuting inference rules in (cut-free) proofs, it is easy to prove the following lemma.

Lemma 7.7. If a sequent has an $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$ proof, it was a P/N-proof.

Given the structure of P/N-proofs, the following lemma has a direct proof.

Lemma 7.8. If a reduced sequent has a $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$ proof in which the last inference rule is the introduction of a positive connective, then it has a positive region proof.

Ketonen [31] used cut admissibility in Gentzen's LK proof system to provide elegant proofs of the invertibility for some introduction rules. His arguments can be used to show the invertibility of the introduction rules for the (negative) connectives $\bot$, $⅋$, $\top$, $\&$, and $\forall$ in $\bar{\bar{\mu}}$MALL. Since we do not have a cut-admissibility result for $\bar{\bar{\mu}}\mathrm{LK}_p$, we will need to make a simple variation on his proof to prove similar statements for $\bar{\bar{\mu}}\mathrm{LK}_p$.

Lemma 7.9. Let $\vdash N, \Gamma$ be a sequent where the top-level connective of N is one of the following (negative) connectives: $\neq$, $\bot$, $⅋$, $\top$, $\&$, and $\forall$. If this sequent has a $\bar{\bar{\mu}}\mathrm{LK}_p$ proof, it has a $\bar{\bar{\mu}}\mathrm{LK}_p$ proof in which the last inference rule introduces N.

Proof:
Assume that the $\vdash B_1 \& B_2, \Gamma$ sequent has a $\bar{\bar{\mu}}\mathrm{LK}_p$ proof Ξ. We want to prove that it has a $\bar{\bar{\mu}}\mathrm{LK}_p$ proof in which the last inference rule introduces $B_1 \& B_2$. Consider the following $\bar{\bar{\mu}}\mathrm{LK}_p^+$ proof.

$$\dfrac{\dfrac{\begin{matrix}\Xi\\ \vdash \Gamma, B_1 \& B_2\end{matrix} \quad \dfrac{\dfrac{}{\vdash B_1, \overline{B_1}}\,\mathit{init}}{\vdash B_1, \overline{B_1 \& B_2}}\,\oplus}{\vdash \Gamma, B_1}\,\mathit{cut} \qquad \dfrac{\begin{matrix}\Xi\\ \vdash \Gamma, B_1 \& B_2\end{matrix} \quad \dfrac{\dfrac{}{\vdash B_1, \overline{B_1}}\,\mathit{init}}{\vdash B_1, \overline{B_1 \& B_2}}\,\oplus}{\vdash \Gamma, B_2}\,\mathit{cut}}{\vdash \Gamma, B_1 \& B_2}\,\&$$

Since *init* is admissible in $\bar{\bar{\mu}}$MALL, it is admissible in $\bar{\bar{\mu}}\mathrm{LK}_p$. If we can eliminate the *cut* rules from this proof, we will have a $\bar{\bar{\mu}}\mathrm{LK}_p$ proof in which the last inference rule introduces $B_1 \& B_2$. We can move the *cut* rule upwards to eliminate it in the usual fashion. In this case, the only issue arrives when the last inference rule of Ξ is a contraction on $B_1 \& B_2$. That is, the left premises of the $\&$-introduction rule above is of the form

$$\dfrac{\dfrac{\begin{matrix}\Xi'\\ \vdash \Gamma, B_1 \& B_2, B_1 \& B_2\end{matrix}}{\vdash \Gamma, B_1 \& B_2}\,C \quad \dfrac{\dfrac{}{\vdash B_1, \overline{B_1}}\,\mathit{init}}{\vdash B_1, \overline{B_1 \& B_2}}\,\oplus}{\vdash \Gamma, B_1}\,\mathit{cut}.$$

This derivation can be transformed into the following derivation.

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\Xi'\\ \vdash \Gamma, B_1 \& B_2, B_1 \& B_2\end{matrix} \quad \dfrac{\dfrac{}{\vdash B_1, \overline{B_1}}\,\mathit{init}}{\vdash B_1, \overline{B_1 \& B_2}}\,\oplus}{\vdash \Gamma, B_1, B_1 \& B_2}\,\mathit{cut} \quad \dfrac{\dfrac{}{\vdash B_1, \overline{B_1}}\,\mathit{init}}{\vdash B_1, \overline{B_1 \& B_2}}\,\oplus}{\vdash \Gamma, B_1, B_1}\,\mathit{cut}}{\vdash \Gamma, B_1}\,C$$

This way, the contraction applied to $B_1 \mathbin{\&} B_2$ is transformed into contractions on the subformulas B_1 and B_2. If we replace contraction with weakening, a similar transformation can be done.

$$\dfrac{\dfrac{\begin{matrix}\Xi' \\ \vdash \Gamma\end{matrix}}{\vdash \Gamma, B_1 \mathbin{\&} B_2}\, W \qquad \dfrac{\dfrac{}{\vdash B_1, \overline{B_1}}\,\textit{init}}{\vdash B_1, \overline{B_1 \mathbin{\&} B_2}}\,\oplus}{\vdash \Gamma, B_1}\,\textit{cut} \qquad \Longrightarrow \dfrac{\begin{matrix}\Xi' \\ \vdash \Gamma\end{matrix}}{\vdash \Gamma, B_1}\, W$$

In this way, the weakening applied to $B_1 \mathbin{\&} B_2$ is transformed into weakenings on the subformulas B_1 and B_2.

Similarly, assume that the $\vdash B_1 \parr B_2, \Gamma$ sequent has a $\bar{\bar{\mu}}\mathrm{LK}_p$ proof Ξ and that Ξ ends in a contraction.

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\Xi' \\ \Gamma, B_1 \parr B_2, B_1 \parr B_2\end{matrix}}{\vdash \Gamma, B_1 \parr B_2}\, C \qquad \dfrac{\dfrac{}{\vdash B_1, \overline{B_1}}\,\textit{init} \quad \dfrac{}{\vdash B_2, \overline{B_2}}\,\textit{init}}{\vdash B_1, B_2, \overline{B_1 \parr B_2}}\,\otimes}{\vdash \Gamma, B_1, B_2}\,\textit{cut}}{\vdash \Gamma, B_1 \parr B_2}\,\parr$$

This derivation can be transformed into the following derivation in which contraction and cut have been switched.

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}\Xi' \\ \Gamma, B_1 \parr B_2, B_1 \parr B_2\end{matrix} \qquad \dfrac{\dfrac{}{\vdash B_1, \overline{B_1}}\,\textit{init} \quad \dfrac{}{\vdash B_2, \overline{B_2}}\,\textit{init}}{\vdash B_1, B_2, \overline{B_1 \parr B_2}}\,\otimes}{\Gamma, B_1, B_2, B_1 \parr B_2}\,\textit{cut} \qquad \dfrac{\dfrac{}{\vdash B_1, \overline{B_1}}\,\textit{init} \quad \dfrac{}{\vdash B_2, \overline{B_2}}\,\textit{init}}{\vdash B_1, B_2, \overline{B_1 \parr B_2}}\,\otimes}{\vdash \Gamma, B_1, B_1, B_2, B_2}\,\textit{cut}}{\vdash \Gamma, B_1, B_2}\, C \times 2}{\vdash \Gamma, B_1 \parr B_2}\,\parr$$

Again, the contraction applied to $B_1 \parr B_2$ is transformed into contractions on the subformulas B_1 and B_2.

The remaining cases involving $\neq$, $\bot$, $\top$, and $\forall$ are similar and simpler. In this way, we prove the invertibility of the introduction for all negative connectives except those for ν. □

Note that the invertibility of these negative connectives implies that the contraction and weakening rule does not need to be applied to formulas with such negative connectives at their top level. In this way, Lemma 7.9 allows us to replace contractions on negative formulas by contractions on their subformulas that are either ν-expressions (which we shall deal with next) or on their positive subformulas.

For the purposes of the rest of this section, we generalize the ν rule to incorporate instances of the structural rules applied to a ν-formula.

$$\dfrac{\vdash \Gamma, S_1\vec{t}, \ldots, S_n\vec{t} \quad \vdash BS_1\vec{x}, \overline{S_1\vec{x}} \quad \cdots \quad \vdash BS_n\vec{x}, \overline{S_n\vec{x}}}{\vdash \Gamma, \nu B\vec{t}}\; C\nu_n \quad n \geq 0$$

Since we are working within $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$, the coinvariants $S_1, \ldots, S_n$ are $\mathbf{N}_1$. This rule has $n+1$ premises. If $n = 0$ this rule is the weakening rule for ν-expression; if $n = 1$ this rule is the ν rule; and if $n \geq 2$, it can be justified using $n-1$ contractions. For example, when $n = 2$ the following combination of ν and contraction rules justify the inference rule $C\nu_2$.

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

Let $\bar{\bar{\mu}}\mathrm{LK}^*_p(\mathbf{N}_1)$ be the proof system that results from formally removing the ν rule and adding the rule $C\nu_i$ for every natural number i. Every $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{N}_1)$ proof can be converted to a $\bar{\bar{\mu}}\mathrm{LK}^*_p(\mathbf{N}_1)$ proof by simply renaming the ν rule as $C\nu_1$.

Given that the side formulas Γ in the $C\nu$ rule appear in only one premise, it is easy to show that this rule permute down over any introduction rule for positive connectives. For this reason, we shall assume that the definition (and completeness) of P/N-proofs are extended to include the $C\nu$ rule as a negative rule.

Lemma 7.10. If B is an $\mathbf{N}_2$-formula with a $\bar{\bar{\mu}}\mathrm{LK}^*_p(\mathbf{N}_1)$ proof then it has a $\bar{\bar{\mu}}\mathrm{LK}^*_p(\mathbf{N}_1)$ proof in which the contraction rule is used only with positive formulas.

Proof:
As Lemma 7.9 shows, if one of the negative connectives other than ν appears in a sequent, we can assume that that connective is immediately applied: in particular, a contraction is not used. As the proof of Lemma 7.9 illustrated, contractions on negative formulas can be permuted to be contractions on positive subformulas or integrated into the $C\nu$ rule. □

We now return to Theorem 7.5 and provide it with a proof. Let B be a pointed formula such that $\vdash B$ has a $\bar{\bar{\mu}}\mathrm{LK}^*_p(\mathbf{N}_1)$ proof. We need to show that $\vdash B$ has a $\bar{\bar{\mu}}$MALL proof, and we do that by transforming the $\bar{\bar{\mu}}\mathrm{LK}^*_p(\mathbf{N}_1)$ proof into a proof without contractions and weakenings. We have shown that we can eliminate the use of these two rules with negative formulas, and when they are used with positive formulas, we can move them into a positive region where we know their use is superfluous. More specifically, assume that there is a $\bar{\bar{\mu}}\mathrm{LK}^*_p(\mathbf{N}_1)$ proof of $\vdash B$ that is a P/N-proof (by Lemma 7.7) in which contraction and weakening are not applied to negative formulas (by Lemma 7.10). Let Ξ be a $\bar{\bar{\mu}}\mathrm{LK}^*_p(\mathbf{N}_1)$ proof satisfying these restrictions. Since the conclusion of Ξ, namely $\vdash B$, is a pointed sequent, Lemma 7.3 implies that the premise and conclusion of all introductions of negative connectives are pointed. Now consider a (pointed) sequent $\vdash \Gamma$ that is the premise of the introduction of a negative connective while also being the conclusion of a positive region proof. By Lemma 7.1, we know that there is a pointed subsequent $\vdash \Gamma'$ of $\vdash \Gamma$ that has a $\bar{\bar{\mu}}$MALL proof. The only difference between Γ and Γ' is that the former can have $\mathbf{N}_1$-formulas, not in the latter. We can then take the $\bar{\bar{\mu}}$MALL proof for $\vdash \Gamma'$ and use the admissibility in $\bar{\bar{\mu}}$MALL of weakening for $\mathbf{N}_1$ formulas to provide a $\bar{\bar{\mu}}$MALL proof of $\vdash \Gamma$. The only remaining detail is to prove that the instances of $C\nu$ used in $\bar{\bar{\mu}}\mathrm{LK}^*_p(\mathbf{N}_1)$ proofs are admissible in $\bar{\bar{\mu}}$MALL: this is easy to show by using the admissibility in $\bar{\bar{\mu}}$MALL of the weakening and contraction of $\mathbf{N}_1$-formulas.

8. Related and future work

The main difference between the hierarchy of $\mathbf{P}_n/\mathbf{N}_n$-formulas used in this paper and the familiar classes of formulas in the arithmetic hierarchy (based on quantifier alternations) is the occurrences of fixed points within formulas. In that regard, $\bar{\bar{\mu}}$MALL and $\bar{\bar{\mu}}\mathrm{LK}_p$ are probably more aptly compared to the extension of Peano Arithmetic based on general inductive definitions found in [39].

Circular proof systems for logics with fixed points have received much attention in recent years, especially within the context of linear logic [5, 17, 19] and intuitionistic logic [14]. Such proof systems generally eschew all first-order term structures (along with first-order quantification). They also eschew the use of explicit invariants and use cycles within proofs as an implicit approach to discovering invariants.

Historically speaking, the logic $\bar{\bar{\mu}}$MALL was developed along with the construction of the Bedwyr model checker [6]. Although that model checker was designed to prove judgments in classical logic, it became clear that only linear logic principles were needed to describe most of its behaviors. The paper [27] illustrates how $\bar{\bar{\mu}}$MALL and its (partial) implementation in Bedwyr can be used to determine standard model-checking problems such as reachability and simulation. A small theorem-proving implementation based on the focused proof system $\bar{\bar{\mu}}$MALL is described in [8]: that prover was capable of proving automatically many of the theorems related to establishing determinacy and totality of $\mathbf{P}_1$ relational specifications.

As mentioned above, whether or not $\bar{\bar{\mu}}\mathrm{LK}_p$ satisfies a cut-elimination theorem or has a (relatively) complete focused proof system are open questions. Resolving both of these questions is an important research problem to consider next.

When we know that the rules for contraction and induction are not involved (as in Section 4), then proof search in $\bar{\bar{\mu}}$MALL resembles computation in the logic programming setting (*i.e.*, involving unification and nondeterministic search). In $\bar{\bar{\mu}}$MALL (in contrast to $\bar{\bar{\mu}}\mathrm{LK}_p$), contraction is not available, leaving the generation of coinvariants as the key feature to concentrate on for automation. A potentially valuable application of our work on $\bar{\bar{\mu}}$MALL is to structure a theorem prover so that the cleverness involved with discovering induction coinvariants could instead be placed on the discovery of lemmas. In particular, one could always choose to use the "obvious coinvariant" when attempting a coinductive proof, much as was done in the prover described in [8] (also, reminiscent of the Boyer-Moore theorem prover [10]). The cleverness required to complete the proofs could be transferred to discovering applicable lemmas. If appropriately organized, such a proof would only require the user to supply a sequence of lemmas; all the remaining details, such as case analysis and coinvariant generation, would be automated.

Many of the results in this paper are based on Chapter 3 of the first author's Ph.D. dissertation [34] and the technical report [35].

9. Conclusions

In this paper, we have started exploring $\bar{\bar{\mu}}$MALL as a linearized version of arithmetic in a way similar to using Heyting Arithmetic as a constructive version of arithmetic. In particular, we have considered three different proof systems. The first is $\bar{\bar{\mu}}$MALL, for which a cut-admissibility theorem is known. The other two are natural variants of $\bar{\bar{\mu}}$MALL that introduce into $\bar{\bar{\mu}}$MALL the rules of contraction and weakening, yielding $\bar{\bar{\mu}}\mathrm{LK}_p$, as well as cut, yielding $\bar{\bar{\mu}}\mathrm{LK}_p^+$. We demonstrate that the third proof system is consistent and powerful enough to encompass all Peano Arithmetic. While it is known that $\bar{\bar{\mu}}$MALL can prove the totality of primitive recursive function specifications, we demonstrate that the non-primitive recursive Ackermann function can also be proved total in $\bar{\bar{\mu}}$MALL. We have also demonstrated that if we can prove in $\bar{\bar{\mu}}\mathrm{LK}_p$ that a certain $\mathbf{P}_1$ relational specification defines a function; then a simple proof search algorithm can compute that function using unification and backtracking search. This approach differs from the proof-as-program interpretation of a constructive proof of the totality of a relational specification. We have also shown a few simple cases when $\bar{\bar{\mu}}\mathrm{LK}_p$ is conservative over $\bar{\bar{\mu}}$MALL.

Acknowledgment: We thank the anonymous reviewers of an earlier draft of this paper for their valuable comments. We also thank Anupam Das for several conversations and suggestions that helped shape this paper's overall structure and results.

References

[1] Aït-Kaci H. Warren's Abstract Machine: A Tutorial Reconstruction. Logic Programming Research Reports and Notes. MIT Press, Cambridge, MA, 1991. URL `https://wambook.sourceforge.net/`.

[2] Baelde D. A linear approach to the proof-theory of least and greatest fixed points. Ph.D. thesis, Ecole Polytechnique, 2008. URL `https://arxiv.org/abs/0910.3383`.

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

[4] Baelde D, Chaudhuri K, Gacek A, Miller D, Nadathur G, Tiu A, Wang Y. Abella: A System for Reasoning about Relational Specifications. *J. of Formalized Reasoning*, 2014. **7**(2):1–89. doi: 10.6092/issn.1972-5787/4650.

[5] Baelde D, Doumane A, Kuperberg D, Saurin A. Bouncing Threads for Circular and Non-Wellfounded Proofs: Towards Compositionality with Circular Proofs. In: Baier C, Fisman D (eds.), LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022. ACM, 2022 pp. 63:1–63:13. doi:10.1145/3531130.3533375.

[6] Baelde D, Gacek A, Miller D, Nadathur G, Tiu A. The Bedwyr system for model checking over syntactic expressions. In: Pfenning F (ed.), 21th Conf. on Automated Deduction (CADE), number 4603 in LNAI. Springer, New York, 2007 pp. 391–397. doi:10.1007/978-3-540-73595-3_28. URL `https://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/cade2007.pdf`.

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

[8] Baelde D, Miller D, Snow Z. Focused Inductive Theorem Proving. In: Giesl J, Hähnle R (eds.), Fifth International Joint Conference on Automated Reasoning, number 6173 in LNCS. 2010 pp. 278–292. doi: 10.1007/978-3-642-14203-1_24.

[9] Banâtre JP, Métayer DL. Programming by Multiset Transformation. *Communications of the ACM*, 1993. **36**(1):98–111. doi:10.1145/151233.151242.

[10] Boyer RS, Moore JS. A Computational Logic. Academic Press, 1979.

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

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

[13] Clark KL. Negation as failure. In: Gallaire J, Minker J (eds.), Logic and Data Bases, pp. 293–322. Plenum Press, New York, 1978. doi:10.1007/978-1-4684-3384-5_11.

[14] Curzi G, Das A. Computational expressivity of (circular) proofs with fixed points. In: 38th Symp. on Logic in Computer Science. 2023 pp. 1–13. doi:10.1109/LICS56636.2023.10175772.

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

[16] Das A, De A, Saurin A. Decision problems for linear logic with least and greatest fixed points. In: 7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022). Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022 pp. 20:1–20:20. doi:10.4230/LIPIcs.FSCD.2022.20.

[17] De A, Jafarrahmani F, Saurin A. Phase Semantics for Linear Logic with Least and Greatest Fixed Points. In: Dawar A, Guruswami V (eds.), 42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022, December 18-20, 2022, IIT Madras, Chennai, India, volume 250 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022 pp. 35:1–35:23. doi:10.4230/LIPICS.FSTTCS.2022.35.

[18] Delande O, Miller D, Saurin A. Proof and refutation in MALL as a game. *Annals of Pure and Applied Logic*, 2010. **161**(5):654–672. doi:10.1016/j.apal.2009.07.017.

[19] Ehrhard T, Jafarrahmani F. 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, 2021 pp. 1–13. doi:10.1109/LICS52264.2021.9470664.

[20] Friedman HM. Classically and Intuitionistically Provably Recursive Functions. In: Müller GH, Scott DS (eds.), Higher Order Set Theory, pp. 21–27. Springer Verlag, Berlin, 1978. doi:10.1007/BFb0103100.

[21] Gehlot V, Gunter C. Normal Process Representatives. In: Mitchell JC (ed.), 5th Symp. on Logic in Computer Science. IEEE Computer Society Press, Philadelphia, Pennsylvania, 1990 pp. 200–207. doi:10.1109/LICS.1990.113746.

[22] Gentzen G. Investigations into Logical Deduction. In: Szabo ME (ed.), The Collected Papers of Gerhard Gentzen, pp. 68–131. North-Holland, Amsterdam, 1935. doi:10.1007/BF01201353. Translation of articles that appeared in 1934-35. Collected papers appeared in 1969.

[23] Gérard U, Miller D. Separating Functional Computation from Relations. In: Goranko V, Dam M (eds.), 26th EACSL Annual Conference on Computer Science Logic (CSL 2017), volume 82 of *LIPIcs*. 2017 pp. 23:1–23:17. doi:10.4230/LIPIcs.CSL.2017.23.

[24] Girard JY. A new constructive logic: classical logic. *Math. Structures in Comp. Science*, 1991. **1**:255–296. doi:10.1017/S0960129500001328.

[25] Girard JY. A Fixpoint Theorem in Linear Logic, 1992. An email posting to linear@cs.stanford.edu archived at `https://www.seas.upenn.edu/~sweirich/types/archive/1992/msg00030.html`.

[26] Grishin VN. Predicate and set-theoretic calculi based on logic without contractions. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, 1981. **45**(1):47–68.

[27] Heath Q, Miller D. A proof theory for model checking. *J. of Automated Reasoning*, 2019. **63**(4):857–885. doi:10.1007/s10817-018-9475-3.

[28] Hodas J, Miller D. Logic Programming in a Fragment of Intuitionistic Linear Logic. *Information and Computation*, 1994. **110**(2):327–365. doi:10.1006/inco.1994.1036.

[29] Horne R, Padovani L. A Logical Account of Subtyping for Session Types. In: Castellani I, Scalas A (eds.), Proc. 14th Workshop on Programming Language Approaches to Concurrency and Communication-cEntric Software, PLACES 2023, volume 378 of *EPTCS*. 2023 pp. 26–37. doi:10.4204/EPTCS.378.3.

[30] Howard WA. The formulae-as-type notion of construction, 1969. In: Seldin JP, Hindley R (eds.), To H. B. Curry: Essays in Combinatory Logic, Lambda Calculus, and Formalism, pp. 479–490. Academic Press, New York, 1980.

[31] Ketonen O. Investigations into the Predicate Calculus. College Publications, 2022. Ed. by S. Negri and J. von Plato.

[32] Kobayashi N, Yonezawa A. Asynchronous communication model based on linear logic. *Formal Aspects of Computing*, 1995. **7**(2):113–149. doi:10.1007/BF01211602.

[33] Liang C, Miller D. Focusing Gentzen's LK proof system. In: Piecha T, Wehmeier K (eds.), Peter Schroeder-Heister on Proof-Theoretic Semantics, Outstanding Contributions to Logic, pp. 275–313. Springer, 2024. doi:10.1007/978-3-031-50981-0_9.

[34] Manighetti M. Developing proof theory for proof exchange. Ph.D. thesis, Institut Polytechnique de Paris, 2022. URL `https://theses.hal.science/tel-04289251`.

[35] Manighetti M, Miller D. Peano Arithmetic and muMALL. Technical Report arXiv:2312.13634, arXiv, 2023. URL `https://arxiv.org/abs/2312.13634`.

[36] Miller D. The π-calculus as a theory in linear logic: Preliminary results. In: Lamma E, Mello P (eds.), 3rd Workshop on Extensions to Logic Programming, number 660 in LNCS. Springer, Bologna, Italy, 1993 pp. 242–265. doi:10.1007/3-540-56454-3_13.

[37] Miller D. Forum: A Multiple-Conclusion Specification Logic. *Theoretical Computer Science*, 1996. **165**(1):201–232. doi:10.1016/0304-3975(96)00045-X.

[38] Miller D, Viel A. Proof search when equality is a logical connective. *Annals of Mathematics and Artificial Intelligence*, 2022. **90**(5):523–535. doi:10.1007/s10472-021-09764-0. *Special Issue on Theoretical and Practical Aspects of Unification*.

[39] Möllerfeld M. Generalized inductive definitions. Ph.D. thesis, University of Münster, 2002. URL `https://nbn-resolving.de/urn:nbn:de:hbz:6-85659549572`.

[40] Paris JB, Kerby LAS. Σ_n-Collection Schemas in Arithmetic. In: Macintyre A, Pacholski L, Paris J (eds.), Logic Colloquium '77, volume 96 of *Studies in logic and the foundations of mathematics*. North-Holland, 1978 pp. 199–209.

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

[42] Simpson A. Cyclic Arithmetic Is Equivalent to Peano Arithmetic. In: Esparza J, Murawski AS (eds.), Foundations of Software Science and Computation Structures - 20th International Conference, FoSSaCS, volume 10203 of *LNCS*. 2017 pp. 283–300. doi:10.1007/978-3-662-54458-7_17.

[43] Stickel M. A PROLOG Technology Theorem Prover: Implementation by an extended PROLOG compiler. *Journal of Automated Reasoning*, 1988. **4**(4):353–380.

[44] Tärnlund SA. Horn Clause Computability. *BIT*, 1977. **17**:215–226. doi:10.1007/BF01932293.