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

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 a use $\bar{\bar{\mu}}$MALL 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 logical connectives 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 $\bar{\bar{\mu}}$MALL relational specifications can be computed using a simple proof search algorithm. By incorporating weakening and contraction into $\bar{\bar{\mu}}$MALL, we obtain $\bar{\bar{\mu}}\mathrm{LK}_p^+$, a natural candidate for a classical sequent calculus for arithmetic. While important proof theory results are still lacking for $\bar{\bar{\mu}}\mathrm{LK}_p^+$ (including cut-elimination and the completeness of focusing), we prove that $\bar{\bar{\mu}}\mathrm{LK}_p^+$ is consistent and that it contains Peano arithmetic. We also prove two conservativity results regarding $\bar{\bar{\mu}}\mathrm{LK}_p^+$ over $\bar{\bar{\mu}}$MALL.

1. Introduction

A feature of first-order logic is not only the presence of propositional connectives, first-order quantifiers, and first-order terms but also the class of non-logical constants called *predicates* that denote relations between terms. When we move from first-order logic to first-order arithmetic, we introduce induction principles and banish undefined predicates 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 propose to study arithmetic based instead on linear logic: in particular, we shall use the $\bar{\bar{\mu}}$MALL logic of [1–3] to capture this linearized version of arithmetic.

Linear logic has played various roles in computational logic. Many applications rely on the ability of linear logic to capture the multiset rewriting paradigm, which, in turn, can encode Petri nets [4], process calculi [5, 6], and stateful computations [7, 8]. 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. Our approach to linearized arithmetic is similar to that expressed recently by Girard [9] about linear logic: "*Linear* logic is an unfortunate expression that suggests a particular system, while it is the key permitting the abandonment of all systems."[1] Here, we propose linearized arithmetic not to have a new, 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 is in contrast to uses of the propositional subset of $\bar{\bar{\mu}}$MALL as a typing systems (see, for example, [10]). 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., [12, 13]).

Our first step with the analysis of arithmetic starts by demonstrating that functions defined using relational specifications in $\bar{\bar{\mu}}$MALL can be computed directly from that relational specification using unification and backtracking search (Section 4). We also introduce a new proof system $\bar{\bar{\mu}}\mathrm{LK}_p^+$ that is identical to $\bar{\bar{\mu}}$MALL except for the addition of weakening and contraction. While adding weakening and contraction to MALL provides a natural and well studied proof system for classical logic, the exact nature of $\bar{\bar{\mu}}\mathrm{LK}_p^+$ is not well understood yet. In particular, for example, if we denote by $\bar{\bar{\mu}}\mathrm{LK}_p$ the result of removing the cut-rule from $\bar{\bar{\mu}}\mathrm{LK}_p^+$, we do not know if $\bar{\bar{\mu}}\mathrm{LK}_p$ and $\bar{\bar{\mu}}\mathrm{LK}_p^+$ prove the same sequents, and we do not know if the completeness of focusing holds for $\bar{\bar{\mu}}\mathrm{LK}_p$. In this paper, we show that $\bar{\bar{\mu}}\mathrm{LK}_p^+$ is consistent, that it contains Peano arithmetic (Section 6), and that $\bar{\bar{\mu}}\mathrm{LK}_p$ has certain conservativity results with respect to $\bar{\bar{\mu}}$MALL (Section 7).

2. Terms and formulas

We use Church's approach [14] to defining 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). For the scope of this paper, we assume that there is a second primitive type ι and that the (ambient) signature $\mathbf{P}_0$ 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.

[1]The original French: "Logique *linéaire* est une expression malvenue qui suggère un système alors qu'il s'agit de la clef permettant de les abandonner tous."

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 since that 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 symbols: we do not admit any non-logical symbols of type $\iota \to \cdots \to \iota \to o$. As a result, there are no atomic formulas, usually defined as formulas with a non-logical symbol as their head. Equality, non-equality, and the fixed point operators are treated as logical connectives since they will all receive introduction rules in the sequent calculus proof systems that we introduce soon.

2.2. Propositional connectives of linear logic

The eight linear logic connectives for MALL are the following.

	conjunction	true	disjunction	false
multiplicative	$\otimes$	1	$\parr$	$\bot$
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 [1].[2] 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 $\overline{B} \parr C$. 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 $[\lambda p \lambda x.x \neq z\ \&\ \forall y.x \neq (s\ (s\ y)) \parr p\ y]$.

2.3. Polarized and unpolarized formulas

The connectives of linear logic are given a *polarity* as follows. The *negative* connectives are $\parr$, $\bot$, $\&$, $\top$, $\forall$, $\neq$, and ν while their De Morgan duals—namely, $\otimes$, 1, $\oplus$, 0, $\exists$, $=$, and μ—are positive. A

[2]This logic was named simply μMALL in [1] but that name is often used to denote the subset of that logic without $=$, $\neq$, $\forall$, and $\exists$: see, for example, [15, 16]. As a result, we have renamed this logic here with the addition of the equality sign to stress our interest in the structure of first-order terms.

$\bar{\bar{\mu}}$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 $\wedge$, *tt*, $\vee$, *ff*, $=$, $\neq$, $\forall$, $\exists$, μ, and ν. Note that 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 $\parr$ 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 $\perp$ 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 expression, such as $((\mu\lambda P\lambda x(B\ P\ x))\ t)$, introduce variables of predicate type (here, P). In the case of the μ fixed point, any expression built using that predicate variable will be considered positively polarized. If the ν operator is used instead, any expressions built using the predicate variables it introduces are considered negatively polarized.

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 positive subformula occurrence is a $\mathbf{P}_n$-formula. Similarly, the $\mathbf{P}_{n+1}$-formulas are those positive formulas for which every negative subformula occurrence is a $\mathbf{N}_n$-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 a $\mathbf{N}_n$-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 $i \rightarrow o$ 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 [1,3] for polarized formulas is the one-sided sequent calculus proof system given in Figure 1. The variable y in the $\forall$-introduction rule is an *eigenvariable*: it is restricted to not be free in any formula in the conclusion of that rule. The application of a substitution θ to a signature Σ (written $\Sigma\theta$ in the $\neq$ rule in Figure 1) is the signature that results from removing from Σ the variables in the domain of θ and adding back any variable that is free in the range of θ. In the $\neq$-introduction rule, if the terms t and t' are not unifiable, the premise is empty and immediately proves the conclusion.

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

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

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

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

Figure 1. The inference rules for the $\overline{\overline{\mu}}$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 $\overline{\overline{\mu}}$MALL

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

Figure 3. Two structural rules

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 $\beta\eta$-long normal form when presenting sequents. Note that formula expressions such as $B\ S\ \vec{t}$ (see Figure 1) are parsed as $(\cdots((B\ S)t_1)\cdots t_n)$ if $\vec{t}$ is the list of terms $t_1, \ldots, t_n$.

If we were working in a two-sided calculus, the ν-rule in Figure 1 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 one rule for ν yields both coinduction and induction. In general, we shall speak of the higher-order substitution term S used in both of these rules as the *invariant* of that rule (*i.e.*, we will not use the term co-invariant even though that might be more appropriate in some settings).

We make the following observations about this proof system.

1. The 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 $\overline{\overline{\mu}}$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 invariant $S = B(\nu B)$.

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.

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 [17]. 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 $\bar{\bar{\mu}}$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

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

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 $\bar{\bar{\mu}}$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 invariants. Finally, given that first-order Horn clauses can interpret Turing machines [18], 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 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.

$$\begin{gathered}
\textit{nat}\ \mathbf{0} \\
\forall x(\textit{nat}\ x \supset \textit{nat}\ (s\ x))
\end{gathered}$$

$$\begin{gathered}
\forall x(\textit{plus}\ \mathbf{0}\ x\ x) \\
\forall x \forall y \forall u(\textit{plus}\ x\ y\ u \supset \textit{plus}\ (s\ x)\ y\ (s\ u))
\end{gathered}$$

$$\begin{gathered}
\forall x(\textit{mult}\ \mathbf{0}\ x\ \mathbf{0}) \\
\forall x \forall y \forall u \forall w(\textit{mult}\ x\ y\ u \wedge \textit{plus}\ y\ u\ w \supset \textit{mult}\ (s\ x)\ y\ w)
\end{gathered}$$

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.

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

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 instead 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.

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

In a similar fashion, 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 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 this relationship 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 a $\mathbf{N}_2$ 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 [19]. In particular, reachability problems were encoded as $\mathbf{P}_1$ formulas, while non-reachability problems were encoded as $\mathbf{N}_1$ formulas. The paper [19] 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 [20] 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.

$$\begin{array}{lcl} \Phi & ::= & \Phi \,\&\, \Phi \mid \exists x.\Phi \mid \forall x.\Phi \mid \Psi \\ \Psi & ::= & t_1 = t'_1 \multimap \cdots \multimap t_n = t'_n \multimap t_0 = t'_0 \quad (n \geq 0) \end{array}$$

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 [21].

The two principle and major 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.

3.4. Definable exponentials

As Baelde showed in [3], the following definitions

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

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 same 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 [22]. 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 non-deterministic rewriting process that iteratively selects β-redexes for reduction. In most typed λ-calculus systems, all such sequences of rewritings will end in the same normal form, although some sequences of rewrites might be very long, and others can be very short. 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 [14], then this function can be represented via the expression $\lambda x.\iota y.\phi(x, y)$. 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 $\forall u(\overline{\mathit{plus}\ \mathbf{2}\ \mathbf{2}\ u} \parr \mathit{nat}\, u)$ (which incrementally constructs the sum of 2 and 2).

Assume that P is a $\mathbf{P}_1$ predicate expression of type $i \to o$ 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 following outline.

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

Although proof search can contain potentially unbounded branches, we can still use the proof search concepts of unification and non-deterministic search to compute the value within a singleton. We now define a non-deterministic 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 non-deterministically select one of the B_i formulas: for the sake of simplicity, assume that we have selected B_1. We define the transition $S \Rightarrow S'$ of state S to state S' by a case analysis of the top-level structure of B_1.

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

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

Lemma 4.1. 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 with value t such that $P\ t$ has a $\bar{\bar{\mu}}$MALL proof.

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

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

Clearly, if we strike out the augmented items (in red), we are left with a regular state. Given that we have a $\bar{\mu}$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 non-deterministic algorithm to a success state with value t. Given the augmented state $\langle \Sigma \mid \theta;\ B_1 \mid \Xi_1, \ldots, B_m \mid \Xi_m;\ s\rangle$, we consider selecting the first pair $B_1 \mid \Xi_1$ and consider the structure of B_1.

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

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 $\bar{\mu}$MALL proof.

Proof:
Given a (cut-free) $\bar{\bar{\mu}}$MALL proof of $\exists! y.Py$, that proof contains a $\bar{\bar{\mu}}$MALL proof of $\exists y.Py$. Since this formula is $\mathbf{P}_1$, there is a $\bar{\mu}$MALL proof for $\exists y.Py$. The forward direction is immediate: given a sequence of transitions from the initial state $\langle y\,;\ P\ y\,;\ y\rangle$ to the success state $\langle \cdot\,;\ \cdot\,;\ t\rangle$, it is easy to build a $\bar{\mu}$MALL proof of $P\ t$. Conversely, assume that there is a $\bar{\bar{\mu}}$MALL proof of $P\ t$ for some term t. By conservativity, there is a $\bar{\mu}$MALL proof of $P\ t$ and, hence, of $\exists y.P\ y$. By Lemma 4.1, there is a sequence of transitions from initial state $\langle y\,;\ P\ y\,;\ y\rangle$ to the success state $\langle \cdot\,;\ \cdot\,;\ s\rangle$, where $P\ s$ has a $\bar{\mu}$MALL proof. Given that the three formulas Pt and Ps and $\forall x_1 \forall x_2.P\ x_1 \supset P\ x_2 \supset x_1 = x_2$, the admissibility of cut for $\bar{\bar{\mu}}$MALL allows us to conclude that $t = s$. ☐

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

5. The totality of the Ackermann function

The question of the expressivity of $\bar{\mu}$MALL has been analyzed by Baelde [2, Section 3.5], who provided a lower bound to it by characterizing a subset of $\bar{\mu}$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 fixpoints and provide proofs of the totality of these functions in a similar fashion to what we did for the $plus$ relation. However, Baelde noted that this encoding is insufficient for a computational interpretation of Ackermann's function. We can add that 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 $\bar{\mu}$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* (that the first two argument uniquely determine the third argument) and *totality* (that for every choice of the first two arguments, there exists a value for the third argument). Here we focus on totality since the proof of determinacy is simpler and more straightforward. The formula stating totality that we wish to prove is the following.

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

We will now illustrate a proof of this formula. In doing so, we will highlight the crucial use of the encoded exponentials available in $\bar{\mu}$MALL$(!, ?)$. For greater clarity, we will use $\multimap$ as a shorthand, and we will keep the 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:

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

At this point, we need to use the induction rule twice with the $\overline{nat\ n}$ and $\overline{nat\ m}$ formulas. The invariants we introduce for these inductions will be the places where we exploit the encoding of the exponentials. In the first induction, we use as invariant 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 invariant needs to be contracted on at a later stage of the proof, hence the need for the exponential. The left premise of the ν rule is immediately verified, since the invariant 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 $\bar{\mu}$MALL.[3] 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:

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

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

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

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 remove to be proved. The base case (where we need to prove the invariant for k being 0) is again proved by a routine inspection of the definition of ack. The antecedent part of the invariant can be used directly since it starts with ?.

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

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

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

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

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

The entire reason to be using $\bar{\bar{\mu}}$MALL$(!, ?)$ to state invariants in this proof is to make this contraction possible. Now we can decompose the new invariant, which is 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.

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

The consistency of $\bar{\bar{\mu}}$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 [23] and Schroeder-Heister [24] 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 [25].) The following theorem proves that adding contraction to $\bar{\bar{\mu}}$MALL does not lead to inconsistency.

Theorem 6.1. $\bar{\bar{\mu}}$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). A straightforward induction on the structure of $\bar{\bar{\mu}}$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 $\bar{\bar{\mu}}$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 $\bar{\bar{\mu}}\mathrm{LK}_p^+$. 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 either binary function symbols or as three place relations. Formally, we will avoid introducing the extra constructors $+$ and $\cdot$, and choose to encode addition and multiplications as relations. In particular, these are defined as the fixed point expressions *plus* and *mult* given in Section 3. The relation between these two presentation is such that the equality $x + y = w$ corresponds to $\mathit{plus}\, x\, y\, w$ and the equality $x \cdot y = w$ corresponds to $\mathit{mult}\, x\, y\, w$. 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 [26].

Proof 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. Given the presence of contraction and weakening, it is possible to show the equivalence of all the different polarizations of Q. Furthermore, the translations of the Peano Axioms can all be proved in $\bar{\bar{\mu}}\mathrm{LK}_p^+$. We illustrate just one of these axioms here: a polarization of the translation of the induction scheme is

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

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 the inference rule of modus ponens. $\square$

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 [27].

This result inspires the two conservativity theorem we prove in this section. The following theorem is our first conservativity result of (cut-free) $\bar{\bar{\mu}}\mathrm{LK}_p$ over $\bar{\bar{\mu}}$MALL.

Theorem 7.1. $\bar{\bar{\mu}}\mathrm{LK}_p$ is conservative over $\bar{\bar{\mu}}$MALL for $\mathbf{P}_1$-formulas. That is, if B is a $\mathbf{P}_1$ formula, then $\vdash B$ has a $\bar{\bar{\mu}}\mathrm{LK}_p$ proof if and only if $\vdash B$ has a $\bar{\bar{\mu}}$MALL proof.

Proof:
The converse direction of this theorem is immediate. Now consider a sequent of the form $\vdash B_1, \ldots, B_n$ that contains only formulas from $\mathbf{P}_1$. A straightforward argument shows that if this sequent has a $\bar{\bar{\mu}}\mathrm{LK}_p$ proof then there is an $i \in \{1, \ldots, n\}$ so that $\vdash B_i$ has a $\bar{\bar{\mu}}$MALL proof. $\square$

Our next conservativity result requires restricting the complexity of invariants used in the induction rule ν. We say that a sequent has a $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{P}_1)$ proof if it has a $\bar{\bar{\mu}}\mathrm{LK}_p$ proof in which all invariants of the proof are $\mathbf{P}_1$. This restriction on proofs is similar to the restriction that yields the $I\Sigma_1$ fragment of Peano Arithmetic [28].

Theorem 7.2. $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{P}_1)$ is conservative over $\bar{\bar{\mu}}$MALL for $\mathbf{N}_2$-formulas. That is, if B is a $\mathbf{N}_2$-formula such that $\vdash B$ has a $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{P}_1)$ proof, then $\vdash B$ has a $\bar{\bar{\mu}}$MALL proof.

The proof of this result would be aided greatly 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 [1, 3] and add contraction and weakening in the usual fashion, we have a natural candidate for a focused proof system for $\bar{\bar{\mu}}\mathrm{LK}_p$. However, the completeness of that proof system is currently open. As Girard points out in [29], the completeness of such a focused (cut-free) proof system would allow the extraction of the constructive content of classical Π_2^0 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 in order to prove Theorem 7.2.

We find it convenient to introduce the following inference rule.

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

This rule is justified by the following combination of ν and contraction rules.

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

Since we are working within $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{P}_1)$, the invariants S and U are $\mathbf{P}_1$.

If Γ_1 and Γ_2 are reduced sequents, we say that Γ_1 *contains* Γ_2 if Γ_2 is a sub-multiset of Γ_1. A *reduced sequent* is a sequent that contains only $\mathbf{N}_1$, $\mathbf{P}_1$, and $\mathbf{N}_2$ formulas. We say that a reduced sequent is a *pointed* sequent if it contains exactly one formula that is either $\mathbf{P}_1$ or $\mathbf{N}_2$.

Finally, we need two additional notions. A *positive region* is a (cut-free) $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{P}_1)$ proof that contains only the inference rules $\mu\nu$, contraction, weakening, and the introduction rules for the positive connectives. A *negative region* is a (cut-free) $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{P}_1)$ *possibly partial* proof in which the open premises are all reduced sequent and where the only inference rules are introductions for negative connectives plus the $C\nu\nu$ rule.

Lemma 7.3. If a reduced sequent Γ has a positive region proof then Γ contains Γ' such that Γ' is a pointed sequent and has a $\bar{\bar{\mu}}$MALL proof.

Proof:
This proof is a simple generalization of the proof of Theorem 7.1. □

Lemma 7.4. If every premise of a negative region contains a pointed sequent with a $\bar{\bar{\mu}}$MALL proof, then the conclusion of the negative region contains a pointed sequent with a $\bar{\bar{\mu}}$MALL proof.

Proof:
This proof is by induction on the height of the negative region. The most interesting case to examine is the one where the last inference rule of the negative region is the $C\nu\nu$ rule. Referring to the inference rule displayed above, the inductive hypothesis ensures that the reduced sequent $\vdash \Gamma, S\vec{t}, U\vec{t}$ contains a pointed sequent Δ, C where Δ is a multiset of $\mathbf{N}_1$ formulas in Γ and where the formula C (that is either $\mathbf{P}_1$ or $\mathbf{N}_2$) is either a member of Γ or is equal to either $S\vec{t}$ or $U\vec{t}$. In the first case, Δ, C is also contained in the endsequent $\Gamma, \nu B\vec{t}$. In the second case, we have one of the following proofs:

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

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

Lemma 7.5. If the reduced sequent Γ has a cut-free $\bar{\bar{\mu}}\mathrm{LK}_p(\mathbf{P}_1)$ proof then Γ has a proof that can be divided into a negative region that proves Γ in which all its premises have positive region proofs.

Proof:
This lemma is proved by appealing to the permutation of inference rules. As shown in [3], the introduction rules for negative connectives permute down over all inference rules in $\bar{\bar{\mu}}$MALL. Not considered in that paper is how such negative introduction rules permute down over contractions. It is easy to check that such permutations do, in fact, happen except in the case of the ν rule. In general, contractions below a ν rule will not permute upwards, and, as a result, the negative region is designed to include the $C\nu\nu$ rule (where contraction is stuck with the ν rule). As a result, negative rules (including $C\nu\nu$) permute down while contraction and introductions of positive connectives permute upward. This gives rise to the two-region proof structure. □

By combining the results above, we have a proof of Theorem 7.2. Note that when a relational specification is given by a $\mathbf{P}_1$ specification, the *totality* and *determinacy* formulas related to that specification (see Section 4) are $\mathbf{N}_2$ formulas. The proof of the totality of the Ackermann function discussed in Section 5 goes beyond Theorem 7.2 by using complex invariants that include universal quantifiers, implications, and the fixed point definitions for the exponential. We can also strengthen Theorem 4.2 to be a theorem about provability in $\bar{\bar{\mu}}\mathrm{LK}_p$ instead of in $\bar{\bar{\mu}}$MALL.

Theorem 7.6. Assume that P is a $\mathbf{P}_1$ expression of type $i \to o$ and that $\exists! y.Py$ has a $\bar{\bar{\mu}}\mathrm{LK}_p$ 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 $\bar{\bar{\mu}}\mathrm{LK}_p$ proof.

Proof:
Given a (cut-free) $\bar{\bar{\mu}}\mathrm{LK}_p$ proof of $\exists! y.Py$, that proof contains a $\bar{\bar{\mu}}\mathrm{LK}_p$ proof of $\exists y.Py$. Since this formula is $\mathbf{P}_1$, there is a $\bar{\bar{\mu}}$MALL proof for $\exists y.Py$. The result follows using Lemma 4.1 and Theorem 7.1. □

8. Related 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, our principle proof systems $\bar{\bar{\mu}}$MALL and $\bar{\bar{\mu}}\mathrm{LK}_p$ are probably more easily related to the extension of Peano Arithmetic based on general inductive definitions in [30].

Circular proof systems for logics with fixed points have received much attention in recent years, especially within the context of linear logic [10, 31, 32] and intuitionistic logic [33]. 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.

The logic $\bar{\bar{\mu}}$MALL was developed along with the construction of the Bedwyr model checker [34]. 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 [19] 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 [35]: that prover was capable of proving automatically many of the theorems related to establishing determinacy and totality of $\mathbf{P}_1$ relational specifications.

Most of the results here are taken from Chapter 3 of the first author's Ph.D. dissertation [36].

9. Conclusions

In this paper, we have started to explore $\bar{\bar{\mu}}$MALL as a linearized version of arithmetic in a way that is 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 of 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 totality of a relational specification.

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 the overall structure and results.

References

[1] 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.

[2] Baelde D. A linear approach to the proof-theory of least and greatest fixed points. Ph.D. thesis, Ecole Polytechnique, 2008. URL `http://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] Gehlot V, Gunter C. Normal Process Representatives. In: 5th Symp. on Logic in Computer Science. IEEE Computer Society Press, Philadelphia, Pennsylvania, 1990 pp. 200–207. doi:10.1109/LICS.1990.113746.

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

[6] 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.

[7] 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.

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

[9] Girard JY. Schrödinger's cut: La logique à la lumière du quantique, 2021. Unpublished., URL `http://girard.perso.math.cnrs.fr/chat.pdf`.

[10] 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.

[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] 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.

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

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

[15] 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 doi:10.4230/LIPIcs.FSCD.2022.20.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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*.

[22] 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.

[23] 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`.

[24] 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.

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

[26] 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.

[27] 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.

[28] 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.

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

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

[31] 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.

[32] 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.

[33] 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.

[34] 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 `http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/cade2007.pdf`.

[35] 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.

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