Abstract

Peano Arithmetic is based on classical logic. In this paper, we present a version of arithmetic based instead on linear logic. This version of arithmetic is called $\mu\text{MALL}$ and we use it to characterize several topics within computational logic. In particular, we show that $\mu\text{MALL}$ directly encodes Horn-clause logic programming and various model checking problems. We also show how the proof search interpretation of $\mu\text{MALL}$ can be used to compute general recursive functions. Finally, we also identify several situations where Peano Arithmetic is conservative over $\mu\text{MALL}$.

2012 ACM Subject Classification Theory of computation \rightarrow Proof theory; Theory of computation \rightarrow Logic and verification

Keywords and phrases Computational logic, linear logic, arithmetic, proof theory

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 usually called predicates that stand for relations between terms. When we move from first-order logic to first-order arithmetic, we banish undefined predicates and instead provide a mechanism to formally define relations between terms. The definition mechanism we use in this paper allows for both least and greatest fixed points. When moving from classical logic to arithmetic, one arrives at a presentation of Peano Arithmetic. In this paper, we consider moving from linear logic to a linearized version of arithmetic which will be formally defined in Section 3 as $\mu\text{MALL}$.

Since one of the best ways to treat linear and classical logic provability is via the sequent calculus, we use that style of proof to present arithmetic. In particular, we start by using an approach to equality and (generic) fixed points proposed independently by Girard [9] and Schroeder-Heister [23] and then extended to treat least and greatest fixed points in a series of more recent papers [3, 6, 16, 24].

Linear logic has played various roles in computational logic. Many such applications rely on the ability of linear logic to capture the multiset rewriting paradigm, which, in turn, allows direct encoding of Petri nets [7], process calculus [14, 19], and stateful computations [13, 20]. Our use of linear logic here will have none of that flavor. While the sequent calculus we use is based on multisets of formulas, we shall not model computation as some dynamics of multisets of tokens. In contrast, when we use linear logic connectives within arithmetic, we are capturing computation and deduction via familiar means relying on relations on numerical expressions. Our approach to linearized arithmetic is similar to that expressed recently by Girard [10] 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 reasoning in arithmetic broadly understood.

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.”
A linear logic approach to arithmetic

Our project here is to propose that MALL, the multiplicative additive subset of linear logic, extended with first-order quantification, term equality, and least and greatest fixed points is an interesting and important setting for studying arithmetic and computational logic more generally. We will argue that linearized arithmetic provides an interesting perspective on not only arithmetic but also logic programming using Horn clauses (Prolog) and many aspects of model theory.

2 Terms and formulas

We use Church’s approach to defining terms, formulas, and abstractions over these by making them all simply typed $\lambda$-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 $\iota$ and that the signature $\Sigma_0$ contains the constructors $z : \iota$ (zero) and $s : \iota \rightarrow \iota$ (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 $\iota$

We first present the logical connectives that relate to first-order structures. The two quantifiers $\forall$ and $\exists$ are both given the type $(\iota \rightarrow o) \rightarrow o$. Equality $=$ and inequality $\neq$ are both of the type $\iota \rightarrow \iota \rightarrow o$.

For $n \geq 0$, the least fixed point operator of arity $n$ is written as $\mu_n$ and the greatest fixed point operator of $n$-ary is written as $\nu_n$, and they both have the type $(\iota \rightarrow \cdots \rightarrow \iota \rightarrow o) \rightarrow \iota \rightarrow \cdots \rightarrow \iota \rightarrow o$, where there are $2n$ occurrences of $\iota$. We seldom write explicitly the arity of fixed points since that can usually be determined from context when its value is important. The De Morgan dual of $\mu$ is $\nu$ and of $=$ is $\neq$.

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

2.2 Propositional connectives

The eight linear logic connectives for MALL are the following.

<table>
<thead>
<tr>
<th></th>
<th>conjunction</th>
<th>true</th>
<th>disjunction</th>
<th>false</th>
</tr>
</thead>
<tbody>
<tr>
<td>multiplicative</td>
<td>$\otimes$</td>
<td>$\bot$</td>
<td>$\top$</td>
<td>$\perp$</td>
</tr>
<tr>
<td>additive</td>
<td>$&amp;$</td>
<td>$\top$</td>
<td>$\vee$</td>
<td>$0$</td>
</tr>
</tbody>
</table>

The four binary connectives have type $\iota \rightarrow o \rightarrow o$ and the four units have type $o$. Formulas involving the set of logical connectives in Section 2.1 and these propositional connectives will be called $\mu$\textsc{MALL} formulas. This mixture of MALL connectives with the treatment of first-order structures given by first-order quantifiers, equality, and fixed points was presented in [4] and many of its proof-theoretic properties were established in [2, 3]. (The use of 0 and 1 as logical connectives is unfortunate for a paper about arithmetic. When shall write numerals in boldface.)

Negation is not a logical connective: instead, we write $\overline{B}$ we mean the formula that results from taking the De Morgan dual of $B$. We occasionally use the linear implication $B \rightarrow C$ as an abbreviation for $\overline{B} \otimes C$.

The connectives of linear logic are given a polarity as follows. The connectives $\otimes, \perp, \vee, \top, \neq$, and $\forall$ are all negative while their De Morgan duals are all positive. A $\mu$\textsc{MALL} formula is positive or negative depending only on the polarity of its top-most connective. Note that the polarity flips between $B$ and $\overline{B}$. A formula is purely positive (resp, purely negative) if every logical connective it contains is positive (resp, negative). Clearly, if $N$ is purely negative then $\overline{N}$ is purely positive.
Unpolarized formulas

We also call these formulas µ version of the classical formula θ = mgu(s, t) if every occurrence of & and ⊕ in θ is replaced by ∧ and ⊕ in Q, every occurrence of ∨ and ⊗ in θ is replaced by ∨ in Q, every occurrence of 1 and ∨ in θ is replaced by tt in Q, and every occurrence of ⊥ in θ is replaced by ⊥ in Q. Notice that if Q has n-occurrences of propositional connectives, then there are 2^n formulas Q which are polarized versions of Q. We never mix polarized and unpolarized connectives in the same formulas.

Figure 1 Inference rules for µMALL.

3 Sequent calculus proof systems for µMALL and µLK

The one-sided sequent calculus proof system for µMALL is given in Figure 1 and comes from [4]. We make the following observations about this proof system.

1. The general form of the initial rule, namely, that the sequent ⊢ Q, Q is provable, is admissible.
2. The µν rule is a specific instance of the more general form of the initial rule.
3. The rule for µ is often called the unfolding rule. This rule captures, in part, the identification of µB with B(µB); that is, that µB is a fixed point of B.
4. The cut rule can be eliminated from this proof system [3].
5. The following rule, which simply unfolds ν-expression, is also admissible in this proof system by using the ν-rule with the invariable S = B(νB).

Provability in MALL is decidable. In a desire to move from MALL to a more general logic, Girard added to MALL the exponentials !, ? to yield full linear logic. While linear logic does not allow for the general application of the structural rules of weakening and contraction, it does allow for these rules to be applied to certain instances of formulas labeled by these exponentials. By this controlled reinserterion of contraction, provability in linear logic becomes undecidable. The µMALL proof system takes a different approach to permitting unbounded behaviors to be encoded: the inference rules of unfolding allow the one occurrence of B in (µB) to be expanded to two occurrences of B in B(µB). In this way, unbounded behaviors can appear in µMALL where it did not occur in MALL. The exponentials of linear logic have no role to play in this paper.

Arithmetic formulas written using classical logic connectives are built from the same connectives presented in Section 2.1 along with the connectives & tt, ∨, ⊥ (instead of the MALL connectives given in Section 2.2). For convenience, we will occasionally allow implications in classical formulas: in those cases, we treat P ⊃ Q as P ⊃ Q where P is the negation normal form of the negation of P.

We also call these formulas unpolarized formulas and we say that a µMALL formula Q is a polarized version of the classical formula Q if every occurrence of & and ⊕ in Q is replaced by ∧ in Q, every occurrence of ∨ and ⊗ in Q is replaced by ∨ in Q, every occurrence of 1 and ∨ in Q is replaced by tt in Q, and every occurrence of ⊥ in Q is replaced by ⊥ in Q. Notice that if Q has n-occurrences of propositional connectives, then there are 2^n formulas Q which are polarized versions of Q. We never mix polarized and unpolarized connectives in the same formulas.
The proof system $\mu$LK is simply the proof system $\mu$MALL but with the following inference rules for weakening and contraction added.

\[
\begin{align*}
\Gamma, Q, Q & \vdash \Gamma, Q \quad \text{(for weakening)} \\
\Gamma & \vdash \Gamma, Q \quad \text{(for contraction)}
\end{align*}
\]

Since induction in our proof systems is performed as introductions on certain formulas and not on (inductive) types, we introduce the following translation function. We denote by $Q^*$ the result of replacing every occurrence of $\forall$ with $\lambda B. \forall x (\text{nat } x \otimes (Bx))$ and every occurrence of $\exists$ with $\lambda B. \exists x (\text{nat } x \otimes (Bx))$.

We define the following four levels of restriction on proof systems that involve the fixed point operators $\mu$ and $\nu$. These restrictions are based solely on occurrences of the $\mu\nu$ inference rules and the introduction rules for $\mu$ and $\nu$.

- **Level 0**: Remove all occurrences of the $\mu$, $\nu$, and $\mu\nu$ rules.
- **Level 1**: Allow the $\mu$ and unfolding rule for $\nu$ while still not allowing the $\mu\nu$ rule.
- **Level 2**: Allow the $\mu$ and (the full) $\nu$ inference rule: the $\mu\nu$ rule is not available.
- **Level 3**: Allows the full version of the $\mu$, $\nu$, and $\mu\nu$ rules.

We write both $\mu$MALL$^l$ and $\mu$LK$^l$ to denote the proof systems that result from imposing the level $l \in \{0, 1, 2\}$ restriction on $\mu$MALL and $\mu$LK, respectively. In the subsequent sections, we shall see how certain activities in the computational logic use $\mu$MALL proofs that satisfy different level restrictions.

The consistency of these systems can be obtained by translating them to second-order linear logic (LL2). In LL2, we can encode an instance $\mu B$ of the least fix-point of an operator as $\forall S [(! \forall x, BS \equiv S x) \rightarrow S x]$. From the cut-elimination theorem for LL2, the following cut-elimination result follows (see [3]).

▶ **Theorem 1.** Cut elimination holds for $\mu$MALL and $\mu$LK.

## 4 Examples of $\mu$MALL theorems and proofs

The formula $\forall x \forall y [x = y \lor x \neq y]$, which is true in classical logic, has the two polarized versions

- $\forall x \forall y [x = y \otimes x \neq y]$ and $\forall x \forall y [x = y \oplus x \neq y]$.

Only the first of these two polarized formulas is provable in $\mu$MALL although both formulas are provable in $\mu$LK.

It is easy to prove that equality is, for example, an equivalence relation. In particular, the formula $\forall x \forall y \forall w. (x \neq y \otimes y \neq w \otimes x = w)$ is a polarized form of the axiom of transitivity for equality. This formula is proved with the following sequent proof.

\[
\begin{align*}
\vdash & x = x \\
\vdash & x \neq w, x = w \\
\vdash & x \neq y, y \neq w, x = w \\
\vdash & \forall x \forall y \forall w. (x \neq y \otimes y \neq w \otimes x = w) \quad & \forall \times 3
\end{align*}
\]

Two other Peano axioms can also be proved directly in this logic. The two formulas

- $\forall x \forall y [(s x) \neq (s y) \otimes x = y]$ and $\forall x [(s x) \neq z]$
are polarized versions of the axiom that the successor is injective and the axiom that zero is not a successor. They both have simple \( \mu \text{MALL} \) proofs.

For another example, let the sets \( A = \{0, 1\} \) and \( B = \{0, 1, 2\} \) be encoded as \( \lambda x. x = 0 \lor x = 1 \) and \( \lambda x. x = 0 \lor x = 1 \lor x = 2 \). A polarized version of the formula \( \forall x. A x \supset B x \) is

\[
\forall x. [(x \neq 0 \land x \neq 1) \supset (x = 0 \lor x = 1 \lor x = 2)]
\]

and this formula has the following \( \mu \text{MALL} \) proof.

\[
\begin{array}{c}
\vdash 0 = 0 \\
\vdash 1 = 1 \\
\vdash x \neq 0, x \neq 0 & x = 1 & x = 2 \\
\vdash x \neq 0 & x \neq 1, x = 0 & x = 1 & x = 2 \\
\vdash \forall x. [(x \neq 0 \land x \neq 1) \supset (x = 0 \lor x = 1 \lor x = 2)]
\end{array}
\]

Here, the doubled horizontal line indicates that more than one inference rule is applied.

Consider, for example, the following two named fixed point expressions used for identifying natural numbers and for computing the ternary relation of addition.

\[
nat = \mu \lambda n. \lambda n(n = z \oplus \exists n'(n = (s n') \otimes N n'))
\]
\[
plus = \mu \lambda p. \lambda n. \lambda m. \lambda p(n = z \otimes m = p) \supset \exists n' \exists p'(n = (s n') \otimes p = (s p') \otimes P n' m p'))
\]

The Peano arithmetic formula \( \forall n \ [nat n \supset plus n z n] \) can be polarized as the \( \mu \text{MALL} \) formula \( \forall n \ [(nat n) \otimes plus n z n] \) and this formula has the following \( \mu \text{MALL} \) proof.

\[
\begin{array}{c}
\vdash \exists_1 z z z \\
\vdash \exists_2 z z z z, plus \ (s n') \ (s n') \\
\vdash (n \neq z \land \forall n'(n \neq (s n') \otimes plus n' z z n')), plus n z n \land \land \ (nat n) \supset plus n z n \\
\vdash \forall n \ [(nat n) \otimes plus n z n]
\end{array}
\]

Here, the \( \exists_1 \) subproof involves only introduction rules while the \( \exists_2 \) subproof involves also an occurrence of the \( \mu \nu \) rule.

As a final example, consider the following derivation.

\[
\begin{array}{c}
\vdash 2 = (s 1) \\
\vdash 3 = (s 3) \\
\vdash plus 1 2 3 \\
\vdash 2 = (s 1) \otimes 4 = (s 3) \otimes plus 1 2 3 \\
\vdash \exists n' \exists p'(2 = (s n') \otimes 4 = (s p') \otimes plus n' 2 p') \\
\vdash (2 = z \otimes 2 = 4) \otimes \exists n' \exists p'(2 = (s n') \otimes 4 = (s p') \otimes P n' 2 p') \\
\vdash plus 2 2 4 \\
\vdash \exists p, plus 2 2 4 \ p
\end{array}
\]

This derivation can be seen as a partial computation of \( 2 + 2 \): to complete this computation, we must construct a similar subproof verifying that \( 1 + 2 = 3 \). In particular, the witness used to instantiate the final \( \exists p \) is, in fact, that sum. Unfortunately, proof construction in this system does not help us to construct the value of this sum. Instead, the first step in building such a proof bottom-up starts with guessing a value and then checking that it is the correct sum. We return to this issue of computing functions-as-relations in Section 10.

We conclude this section with more detail on the relation with traditional systems of arithmetic. Peano Arithmetic is usually presented as the system consisting of classical logic with equality, and the additional constants \( s, 0, +, \cdot \) and the axioms
A linear logic approach to arithmetic

\[
\forall x (s\ x \neq 0)
\]

\[
\forall x y (s\ x = s\ y) \supset (x = y)
\]

\[
\forall x (x + 0 = x)
\]

\[
\forall x y (x + s\ x = s\ (x + y))
\]

\[
\forall x (s\ x = 0)
\]

\[
\forall x (x \cdot 0 = 0)
\]

\[
\forall x (x \cdot s\ y = (x \cdot y) + x)
\]

\[
\forall x y (x \cdot (y + z) = (x \cdot y) + x)
\]

\[
\forall x \forall y (x = y \supset (x \cdot y = z))
\]

\[
\forall x \forall y (\exists n (n = z) \supset (x \cdot y = z))
\]

We wish to avoid introducing extra symbols and, as a result, we encode addition and multiplications as relations. We can then extend the translation \((\cdot)^{\circ}\) so that it accounts for these differences:

\[
(x + y = z)^{\circ} := \text{plus } x y z
\]

\[
(x \cdot y = z)^{\circ} := \text{mult } x y z
\]

where \(\text{mult} = \mu M \lambda n \lambda m \lambda p ((n = z \otimes p = 0) \oplus \exists n' \exists p' (n = (s\ n') \otimes \text{plus } m\ p' \otimes M\ n'\ m\ p'))\)

The following theorem is straightforward to prove.

\[\text{Theorem 2 (Soundness and completeness of } \mu \text{LK for Peano arithmetic). } \] Let \(Q\) be any arithmetic formula and let \(\hat{Q}\) be a polarized version of \(Q\). Then \(Q\) is provable in Peano arithmetic if and only if \((\hat{Q})^{\circ}\) is provable in \(\mu \text{LK}\).

\[\text{Proof. } \] We showed how some of the translations of Peano Axioms can be proved: the remaining axioms can be done analogously. In particular, a polarization of the translation of the induction scheme is

\[
(A0 \otimes \forall x (\text{nat } x \supset \forall x \forall y (x = y \supset (x \cdot y = z)) \supset \forall x \forall y (\forall x \forall y (x = y \supset (x \cdot y = z)))) \supset \forall x \forall y (\forall x \forall y (x = y \supset (x \cdot y = z)))
\]

By an application of the \(\nu\) rule to the second occurrence of \((\text{nat } x)^{\circ}\) we obtain a proof of this axiom.

Conversely, an instance of the rule \(\nu\) on the fix-point \(\text{nat}\) can be simulated by instantiating the induction scheme on the negated invariant \(\overline{S}\), and then applying cut to this and the premise \(\vdash \Gamma, S\). In the case the induction is performed directly on the \(\text{plus}\) or \(\text{mult}\) predicates, this corresponds to an application of one of their defining axioms right after an induction on their first argument.

\[\text{Proposition 3. } \] The weakening and contraction rules are admissible in \(\mu \text{MALL}\) for purely negative formulas.

5 Linearized arithmetic

We shall first consider \(\mu \text{MALL}\) and \(\mu \text{LK}\) proofs at level 3 restriction (i.e., with no restrictions on the occurrences the \(\mu\) rule and the fixed point rules). We will, however, consider restrictions on the nesting structures of polarities within theorems and invariants.

When trying to compare \(\mu \text{MALL}\) with \(\mu \text{LK}\), we find that there are a number of statements that are provable in \(\mu \text{LK}\) but not in \(\mu \text{MALL}\) since they require contraction to be proved. Take for example the formula \(\perp \rightarrow \perp \otimes \perp\), which is easily provable in \(\mu \text{LK}\) by using contraction, but is not provable in \(\mu \text{MALL}\). As mentioned in Section 4, the formula \(\forall x \forall y (x = y \otimes x \neq y)\) is provable in \(\mu \text{LK}\) but not in \(\mu \text{MALL}\). In order to find an interesting fragment where we can compare the expressivity of both systems, we will restrict the polarities of the connectives that appear in a formula. In Section 7, we defined the notions of purely positive and purely negative formulas. The following proposition is well known and can be proved by induction of the structure of negative formulas [3].

\[\text{Proposition 3. } \] The weakening and contraction rules are admissible in \(\mu \text{MALL}\) for purely negative formulas.
Another way to state this proposition is the linear logic equivalence $\text{N} \rightsquigarrow_\text{N}$ for purely negative formulas $\text{N}$. Thus, expressions such as $\text{nat} 5$ and $\text{plus} \ n \ m \ p$ can be used any number of times within a $\mu\text{MALL}$ proof. If we presented a two-sided sequent system for $\mu\text{MALL}$ then assumptions such as $\text{nat} 5$ and $\text{plus} \ n \ m \ p$ can be used any number of times. As a result, $\mu\text{MALL}$ proof can, occasionally, feel like working in a more classical setting.

We can now state a first correspondence between formulas provable classically and in the linearized system: classical arithmetic is conservative over linearized arithmetic for purely positive formulas.

**Theorem 4.** Let $\Gamma$ be a multiset of purely positive formulas. If $\vdash \Gamma$ has a $\mu\text{LK}$ proof, then there exists a $P \in \Gamma$ such that $\vdash P$ has a $\mu\text{MALL}$ proof.

**Proof.** This proof proceeds by structural induction on the structure of $\mu\text{LK}$ proofs. In the base case, since the $\nu\mu$ rule is not applicable, the only possible base cases are the introduction rules for $\equiv$ and $1$, and, in both cases, the theorem holds immediately.

In the inductive step, consider the case of an application of the $\otimes$ rule to a sequent $\vdash \Gamma, \Delta, P \otimes Q$.

Both of the sequents $\vdash \Gamma, P$ and $\vdash \Delta, Q$ above the inference line consist of purely positive formulas, and so by inductive hypothesis each one of them contains a formula that is provable in $\mu\text{MALL}$. We distinguish three different subcases depending on which of these formulas are:

- $P$ and $Q$ are selected. Then by an application of $\otimes$ we can prove in $\mu\text{MALL}$ $P \otimes Q$, which appears in the endsequent.
- $R \in \Gamma$ is selected. Then $R$ is provable in $\mu\text{MALL}$ by inductive hypothesis and appears in the endsequent.
- $R \in \Delta$ is selected. As in the previous point.

The cases for $\oplus$, $\exists$, $\mu$ are treated analogously (and are actually simpler since the context does not need to be split). We are left with the cases of weakening and contraction and in these cases, the conclusion is immediate.

A **bipole** is defined to be a negative formula that no negative connective appears in the scope of a positive connective. Thus, a bipole consists of some negative top-level connectives, with purely positive subformulas underneath.

By employing the polarizing translation from Peano Arithmetic introduced in Section 4, we can view part of the Arithmetical Hierarchy in terms of polarization.

**Proposition 5.** Let $P$ be a formula of Peano Arithmetic, then

- $P$ is $\Sigma_0^1$ if and only if there is a polarization $\hat{P}$ that is purely positive.
- $P$ is $\Pi_2^1$ if and only if there is a polarization $\hat{P}$ that is a bipole.

One should not be led into confusion by thinking that, thanks to this theorem, we could prove open purely positive formulas and then strengthen them by universal quantification in order to get stronger theorems expressing, for example, the totality of a function: indeed a formula such as $\exists x. \text{plus} \ a \ b \ x$ is not provable, since it requires more information on the input variables (and the proof needs to proceed by induction on them). The provable formula that expresses totality of the plus relation is then $\forall x. \text{nat} \ x \supset \forall y. \text{nat} \ y \supset \exists z. \text{plus} \ x \ y \ z$, and it is a bipole. Going beyond bipole formulas seems to be less interesting: indeed we immediately find formulas such as $\bot \rightarrow_0 \bot \otimes \bot$ that are easily provable in classical logic, but not linearly. We should concentrate on bipole formulas for the rest of the section.

Now that induction is available, we should restrict occurrences of inductive invariants to be purely positive in order to prevent more complex formulas from appearing in the proof. We call $\mu\text{LK}_1$ the system consisting of the same rules of $\mu\text{LK}$ but where the inductive invariants are restricted to be purely positive. The notation comes from the fact that this fragment is similar to the fragment $I\Sigma_1$.
of Peano Arithmetic. We will have that this system coincides with \(\mu\text{MALL}\) on bipole statements; in order to show this, we need a preliminary lemma:

**Lemma 6.** Let \(\vdash \Gamma\) be a sequent consisting of a set of \(\nu\)-expressions \(\Delta_1\) and a set of purely positive formulas \(\Delta_2\) and provable in \(\mu\text{LK}_1\). Then there exists a single formula \(P\) in \(\Delta_2\) such that \(\vdash \Delta_1, P\) has a proof where the contraction rule is only applied to \(\nu\)-expressions.

**Proof.** By induction on the structure of the proof of \(\vdash \Gamma\). This fact holds immediately for the three base cases, namely, \(\mu\nu\) and the introduction rules for \(=\) and 1.

In the inductive step, consider the two last inference rule appearing in the proof of \(\Gamma\). We need to show that if the lowest one is a contraction on a purely positive formula \(Q\), then we can permute it above the second-to-last rule, and to exhibit the single provable \(P\).

First, we note that the contraction can be permuted past any rule that does not have \(Q\) as a major premise. Consider the case of induction:

\[
\frac{\vdash \Gamma, Q, Q, S \vec{t}}{\vdash \Gamma, Q, v B \vec{t}}
\]

In the right branch, remember that in the system under consideration \(S\) has to be purely positive, hence \(S \vec{x}\) is purely negative and \(B S \vec{x}\) is purely positive, and the induction hypothesis can be used to get a proof with contractions only on negatives.

In the left branch, we can apply the induction hypothesis to \(\vdash \Gamma, Q, v B \vec{t}\) and distinguish two cases: either we have a proof of \(\vdash \Delta_1, S \vec{t}\), or we have a proof of \(\vdash \Delta_1, Q\) (or possibly another formula from \(\Delta_2\)). In the first case, we can remove the use of contraction thanks to the induction hypothesis, and the endsequent is provable with the induction. In the second case, we can remove the induction altogether.

In the case of the tensor, we have several possibilities depending on how the context is split. For example:

\[
\frac{\vdash \Gamma_1, Q, A \quad \vdash \Gamma_2, Q, B}{\vdash \Gamma_1, \Gamma_2, Q, A \otimes B}
\]

Note that, since the considered formulas are either negative \(\nu\)-expressions or purely positive, both \(A\) and \(B\) need to be positive. By applying the induction hypothesis to both branches, we have that their relevant formula could be either \(Q, A\) (resp. \(B\)) or another positive formula. If it is \(Q\) on both branches, then the sequent \(\vdash \Delta_1, Q\) is provable without contractions on positives. If \(Q\) is not relevant in one of the two branches, then we can remove the use of contraction altogether.

Next, we show that we can permute a contraction past any rule applied to \(P\). Consider the case of the existential rule:

\[
\frac{\vdash \Gamma, Pt_1, Pt_2}{\vdash \Gamma, \exists x P, \exists x P}
\]

Then by induction hypothesis at most one of the two instances is provable, and we can remove the contraction and use only the relevant existential introduction.

**Theorem 7.** Any bipole formula provable in \(\mu\text{LK}_1\) is provable in \(\mu\text{MALL}\).
We call formulas of the form \( \Phi \) when implementing proof search in proof systems containing eigenvariables or term-level bindings with formulas of the form \( \Gamma \). By lemma 6 they are provable by using contraction only on \( \nu \)-expressions. But by lemma 3 contraction is admissible in \( \mu \text{MALL} \) for these formulas, and so the sequents are provable in \( \mu \text{MALL} \).

Much more can be developed between linearize arithmetic, Peano arithmetic, arithmetic hierarchy, and polarization. We shall turn our attentions now, however, to examine some other topics within computational logic.

### 6 Generalized unification problems

The restriction level 0 yields the proof system \( \mu \text{MALL}^0 \) for formulas that do not contain fixed point operators. Such formula are quantified MALL formula only over the judgments for equality and inequality. Such formulas can be considered to be generalized forms of unification problem. More specifically, the first-order unification problem containing the free variables \( x_1, \ldots, x_n \) and the disagreement pairs \( \langle t_1, t'_1 \rangle, \ldots, \langle t_m, t'_m \rangle \) can be encoded as the quantified formulas

\[
\exists x_1 \ldots \exists x_n [t_1 = t'_1 \land \ldots \land t_m = t'_m] \quad (n, m \geq 0).
\]

When implementing proof search in proof systems containing eigenvariables or term-level bindings [21, 22], such unification problems need to be generalized to contain mixed quantification [18]: that is, they are encoded as formulas of the form

\[
Q_1 x_1 \ldots Q_n x_n [t_1 = t'_1 \land \ldots \land t_m = t'_m] \quad (n, m \geq 0)
\]

where \( Q_i \) is either \( \forall \) or \( \exists \). A more expressive extension, however, would allow replacing simple equations with implicational judgments among equations. Consider the following two classes of formulas.

\[
\Phi ::= \Phi \land \Phi \mid \exists x. \Phi \mid \forall x. \Phi \mid \Psi
\]

\[
\Psi ::= t_1 = t'_1 \lor \cdots \lor t_n = t'_n \lor t_0 = t'_0 \quad (n \geq 0)
\]

We call formulas of the form \( \Phi \) generalized unification problems. The inequality of two terms \( t \) and \( t' \) can be encoded as the formula \( t = t' > c = d \) where \( c \) and \( d \) are two, distinct closed formula.

Let \( P_0 \) be the polarization scheme that replaces \( \land \) with \( \otimes \) and \( t_1 = t'_1 \lor \cdots \lor t_n = t'_n \lor t_0 = t'_0 \) with \( t_1 \neq t'_1 \land \cdots \land t_n \neq t'_n \land t_0 = t'_0 \).

**Theorem 8.** Let \( \Phi \) be a generalized unification problem and let \( \Phi \) be the polarization of \( \Psi \) following the \( P_0 \) scheme. Then, \( \Phi \) is provable in \( \mu \text{LK} \) if and only if \( \Phi \) is provable in \( \mu \text{MALL} \).

**Proof.** Since the rules in \( \mu \text{MALL} \) are all admissible in \( \mu \text{LK} \), the converse implication is immediate. To prove the forward direction, we prove the more general statement: If \( \Gamma \) is provable in \( \mu \text{LK} \), where \( \Gamma \) is a multiset of polarized versions of \( \Phi \)-formulas, then there is a formula \( G \in \Gamma \) such that \( \Gamma \vdash G \) is provable in \( \mu \text{LK} \). We proceed by induction on the structure of such a \( \mu \text{LK} \) proof. If the last inference rule was either contraction or weakening, the result is immediate. If the last inference rule is cut, then there must be a side formula in one of the premises that is provable since it cannot be the case that both \( Q \) and \( \overline{Q} \) are provable (follows from the cut-elimination of \( \mu \text{LK} \)). All other cases for the inference rules are immediate except for the \( \exists \) introduction, in which case it is not true in general. Fortunately, the general form of \( \exists \) does not appear here. Instead, the \( \exists \) rule is only applied with formulas of the form \( t_1 \neq t'_1 \land \cdots \land t_n \neq t'_n \land t_0 = t'_0 \). Since the introduction rules for \( \exists \) and \( \forall \) are invertible, the proof follows immediately.
Thus, we have the result that provability in the classical and linear setting coincides: this means that contraction and weakening plays no role in such generalized unification problems. If we consider only unification problems without implications (that is, force $n = 0$ in the description of $\Psi$ above), then provability of $\Phi$ formulas is decidable. However, when implications are permitted, provability of $\Phi$ formulas is undecidable \cite{25}.

### 7 Logic programming and model checking

The restriction level 1 yields the proof system $\mu \text{MALL}^1$ in which it is possible to unfold both $\mu$ and $\nu$ fixed points. The use of invariants are, however, not permitted in such proofs. As we show now, the proof theory of logic programming with Horn clauses is completely described by using $\mu \text{MALL}^1$ proofs of purely positive formulas.

The connection between Horn clauses and least fixed points is well-known and goes back to at least the Clark completion of Horn clauses \cite{5}. To illustrate, consider the following two Horn clauses axiomatizing addition, written using Prolog-style syntax.

\[
\forall N. \quad \text{plus} \; z \; N \; \; N \; := \; \text{tt}
\]

\[
\forall N' \forall M' \forall P'. \quad \text{plus} \; (s \; N) \; (s \; P) \; := \; \text{plus} \; N \; M \; P
\]

By moving the pattern of variables in the head to an equivalent form in the body of these clauses, we have the modified but equivalent clauses

\[
\forall N' \forall M' \forall P'. \quad \text{plus} \; N \; M \; P \; := \; \exists N' \exists P' \; (N = z \land M = P)
\]

\[
\forall N' \forall M' \forall P'. \quad \text{plus} \; N \; M \; P \; := \; \exists N' \exists P' \; (N = (s \; N') \land P = (s \; P') \land \text{plus} \; N \; M \; P).
\]

These two clauses can now be merged into one by introducing a disjunction.

\[
\forall N' \forall M' \forall P', \; \text{plus} \; N \; M \; P \; := \; (N = z \land M = P) \lor \exists N' \exists P' \; (N = (s \; N') \land P = (s \; P') \land \text{plus} \; N \; M \; P).
\]

This latter clause is essentially the same things as the $\mu \text{MALL}$ definition for $\text{plus}$ given in Section 4 using the polarization that replaces all propositional connectives with their positive variants. In general, all Horn clauses can be rewritten in this fashion so that all queries against such encoded Horn clauses are purely positive expressions. An immediate consequence of Theorem 4 is the fact that a classical proof of such queries is also provable in linearized arithmetic. It is also clear that if there exists a $\mu \text{MALL}$ proof of a purely positive formula, then that proof is actually a $\mu \text{MALL}^1$ proof.

Switching to model checking terminology, such logic programming queries address reachability problems. But what about non-reachability? As has been shown \cite{11}, the notion of negation as finite failure can be captured by (in our setting) by building a $\mu \text{MALL}^1$ proof of a purely negative expression. In this setting, the restriction to level 1 proofs is significant since that restriction rules out the use of invariants. Along a similar vein, the model checking problem of determining simulation and bisimulation of two transition systems is easily written as a bipole formula \cite{12, 17} (assuming that the transition systems are defined using purely positive expressions).

In general, capturing simulation with proofs restricted to level 1 requires, however, that the transition systems are acyclic. Capturing both non-reachability and simulation (and non-simulation) are both possible by moving to restriction level 2. For example, consider a graph that may contain cycles and consider a proof that there is no path from, say, node $a$ and node $b$. This is provable about using an invariant $S$ that encodes a connected component containing $a$ but not $b$. The coinductive proof that one must then build must show that the set $S$ is closed under one-step transitions and contains $a$ and does not contain $b$. Such reasoning does not, surprisingly, need to use the rule $\mu \nu$ rule \cite{12}.
Consider the binary relation on natural numbers $\phi(x, y)$. We say that $\phi$ encodes a function from $\mathbb{N}$ to $\mathbb{N}$ if $\phi(x, y)$ holds exactly when $f(x) = y$. One familiar application of proof theory to computation is to extract from a natural deduction proof of $\forall x:\forall y.\phi(x, y)$ a $\lambda$-expression that can be seen of the description of an algorithm for computing $f$. The $\lambda$-term expression must also be interpreted by a rewriting mechanism that non-deterministic selects $\beta$-redexes for reduction. In most typed $\lambda$-calculus systems, any sequence of rewritings will end in the same normal form (strong normalization). Of course, some sequence of rewritings might be very long and others could be very short.

We now describe a different mechanism for computing the function that underlies a relation. This mechanism will not rely on the Curry-Howard correspond nor on $\lambda$-terms. Instead, we rely directly on a proof search mechanism.

One way to prove that a binary relation $\phi$ encodes a function is to prove the totality and determinacy properties of $\phi$: that is, prove

$$[\forall x: \exists y. \phi(x, y)] \land [\forall x: \forall y_1: \forall y_2. \phi(x, y_1) \supset \phi(x, y_2) \supset y_1 = y_2].$$

Clearly, what is at the core of such properties is to prove that the for every natural number $x$, the predicate $\lambda y. \phi(x, y)$ denotes a singleton set. Singleton can play an important role in proofs, especially focused proofs. We write $\exists! x. P$ to denote the formula $(\exists x. P) \land (\forall x_1: x_2. P x_1 \supset P x_2 \supset x_1 = x_2)$.

Note that if $P$ and $Q$ are predicates of arity one and if $P$ denotes a singleton, then $\exists!(P \land Q)$ is equivalent to $\forall x[P x \supset Q x]$ and $\forall x[P x \supset Q x]$ are logically equivalent. (We shall assume here that $P$ is a purely positive expression with $x$ as its only free variable: as we have seen in Section 7, such expressions can capture general recursive predicates.) Notice that the proof search semantics of these equivalent formulas are surprising different. In particular, if we attempt to prove $\forall x[P x \land Q x]$, 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 $\forall x[P x \supset Q x]$ then we simply allocate an eigenvariable $w$ (which we will eventually instantiate with $t$) and then attempt to prove the sequent $P w \supset Q y$. Obviously, such an attempt at building a proof might actually compute the value $t$ (especially if we can restrict proofs to Level 1). Thus, singletons allow for an ambiguity of polarity: the positive formula $\exists!(P x \land Q x)$ can be changed to the negative formula $\forall x[P x \supset Q x]$ and vice versa.

**Example 9.** Given the definition of addition on natural number found in Section 4, it is an easy matter to prove in $\mu$MALL the following totality and determinancy theorems.

$$[\forall x_1: \forall x_2. \text{nat } x_1 \supset \text{nat } x_2 \supset \exists! y. (\text{plus}(x_1, x_2, y) \land \text{nat } y)]$$

$$[\forall x_1: \forall x_2. \text{nat } x_1 \supset \text{nat } x_2 \supset \forall y_1: \forall y_2. \text{plus}(x_1, x_2, y_1) \supset \text{plus}(x_1, x_2, y_2) \supset y_1 = y_2].$$

These proofs require both induction as the $\mu\nu$ version of the initial rule. Using the cut rule with (the obvious) proofs of $\text{nat } 2$ and $\text{nat } 3$, we know that $\lambda y. (\text{plus } 2 3 y)$ denotes a singleton. In order to compute the sole member of the singleton $\lambda y. (\text{plus } 2 3 y)$, we could perform cut-elimination with the inductively proved totality theorem in that example. Instead of such a proof-reduction approach to computation, the proof search approach starts by replacing the goal $\exists! y. (\text{plus } 2 3 y \land \text{nat } y)$ with $\forall y. (\text{plus } 2 3 y \supset \text{nat } y)$. Attempting to prove this second formula leads to an incremental construction of the answer substitution of $y$, namely, $5$.

Assume that $P$ is a purely positive formula and that we have a $\mu$MALL proof that $P$ is a singleton. As we stated above, this means that we have a $\mu$MALL proof of $\forall x[P x \supset \text{nat } x]$. This proof can be understood as a means to compute the unique element of $P$ except that there might be instances of the induction rule used to prove $\exists! x[P x \land \text{nat } x]$ and, hence, $\forall x[P x \supset \text{nat } x]$. If we can force, however, the proof of this latter formula to be restricted to level 1, then such a restricted proof provides an explicit
A linear logic approach to arithmetic

computation. It is, however, not the case that if there is a \( \mu \text{MALL} \) proof of \( \forall x[Px \supset \text{nat } x] \) then it also has a \( \mu \text{MALL}^1 \) proof.

**Example 10.** Let \( P \) be \( \mu(R(\lambda x.x = z \oplus (R(sx)))) \). Clearly, \( P \) denotes the singleton set containing zero. There is also a \( \mu \text{MALL} \) proof of \( \forall x[Px \supset \text{nat } x] \) but there is not (cut-free) \( \mu \text{MALL}^1 \) proof of this theorem since just using unfoldings will lead to an unbounded proof search attempt which roughly follows the following outline.

\[
\vdash P(s(sy)), \text{nat } y \\
\vdash \text{nat } z \\
\vdash P(sy), \text{nat } y \\
\vdash P(y), \text{nat } y
\]

None-the-less, we can still use the proof search concepts of unification and non-deterministic search to provide the means to compute values within a singleton.

We define a non-deterministic algorithm as follows. The state of this algorithm is a sequent-like structure \((x_1, \ldots, x_n; B_1, \ldots, B_m); \text{nat } t)\), where \( t \) is a term, \( B_1, \ldots, B_m \) is a multiset of purely positive formulas, and all variables free in \( t \) and \( B_1, \ldots, B_m \) are in the multiset of variables \( x_1, \ldots, x_n \). A success state is one of the form \((\cdot; \cdot; \text{nat } t)\) (that is, when \( n = m = 0 \)) and that state is said to have value \( t \).

Given the state \( S = (\Sigma; B_1, \ldots, B_m); \text{nat } t) \) with \( m \geq 1 \), we can non-deterministically select one of the \( B_i \) formulas: for sake of simplicity, assume that we have selected \( B_i \). We define the transition to another state, written as \( S \Rightarrow S' \) depending on the top-level structure of \( B_i \).

- If \( B_i \) is \( u = v \) and the terms \( u \) and \( v \) are uninifiable, then we transition to \((\Sigma\theta; B_2\theta, \ldots, B_m\theta); \text{nat } (t\theta))\).
- If \( B_i \) is \( B \otimes B' \) then we transition to \((\Sigma; B, B', B_2, \ldots, B_m); \text{nat } t)\).
- If \( B_i \) is \( B \oplus B' \) then we transition to either \((\Sigma; B, B_2, \ldots, B_m); \text{nat } t)\) or \((\Sigma; B'; B_2, \ldots, B_m); \text{nat } t)\).
- If \( B_i \) is \( \mu B \) then we transition to \((\Sigma; B(\mu B\theta'), B_2, \ldots, B_m); \text{nat } t)\).
- If \( B_i \) is \( \exists y. B \) then we transition to \((\Sigma, y; B y, B_2, \ldots, B_m); \text{nat } t)\) (assuming, of course, that \( y \) is picked to not be in \( \Sigma \)).

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

**Theorem 11.** Assume that \( P \) is a purely positive formula and that we have a \( \mu \text{MALL} \) proof of \( \exists y.P y \; \& \; \text{nat } y : \text{thus }, P \) denotes a singleton set containing \( t \). There is a sequence of transitions from the initial state \((y; P y; \text{nat } y)\) to a success state with value \( t \).

**Proof.** In order to prove this theorem, we define the notion of an augmented state which is a structure of the form \((\Sigma; \theta; B_1| \ E_1, \ldots, B_m| \ E_m); \text{nat } t)\), where \( \theta \) is a substitution that maps the variables in \( \Sigma \) to closed terms and where \( E_i \) is a \( \mu \text{MALL} \) proof of \( B_i\theta \) for all \( i = 1, \ldots, m \). Clearly, if we strike out the augmented items (in red), then we have a regular state. Given that we have a \( \mu \text{MALL} \) proof of \( \exists y.P y \), we know that there must exist a \( \mu \text{MALL} \) proof \( E_0 \) of \( Pt \) for some term \( t \). Note that there is no occurrences of induction in \( E_0 \). Consider the following initial augmented state \((y| \theta_0; P y| \ E_0); \text{nat } y)\), where \( \theta_0 \) is the substitution of \( y \) with \( t \). The augmented material is of two forms. The substitution \( \theta_0 \) accumulates the construction of the final value via composing various smaller and incremental substitutions. The proof structures \( E_i \) provide an oracle that can steer this non-deterministic algorithm to a success state with value \( t \). The notion of transition is lifted in the obvious way to augmented states to augmented states but this time, the information in additional proofs inform which transitions are now legal. Termination of this algorithm is also ensured since the number of occurrences of inference rules in the included proofs decreases at every step of the transition.
Thus, a (naive) proof-search-style algorithm involving both unification and non-deterministic search is sufficient for computing the functions encoded in relations. Since the totality of functions is a bipole formula, we have seen in section 5 that we can capture at least those functions that are provably total in a system analogous to $IE_1$. This is the class of primitive recursive functions, and [2] gives a coding for them in $\mu$MALL.

9 Focused proof systems

Most of the metatheory results described in this paper arise from showing how certain inference rules (such as contraction) permute over other inference rules. Polarity also played a big role in such proofs since, for example, the rules for introducing negative connectives are invertible (and hence permute over all other inference rules).

It is possible to impose a great deal of structure on inference rules by working directly with the polarity of logical connectives. Focus proof system provide just such structure. The first comprehensive focusing proof system was given by Andreoli [1] for linear logic. Focused proof systems have also been given for classical and intuitionistic logic [15].

In the setting of arithmetic, the only known complete focused proof system is the $\mu$MALLF proof system [3, 4] given in Figure 2. In that proof system, we have two kinds of sequents, $\vdash \Delta \uparrow \Gamma$ and $\vdash Q \downarrow \Gamma$, where $\Delta$ and $\Gamma$ are multisets of formulas and $Q$ is a formula. In both of these kinds of sequents, introduction rules only affect formulas that lie between the up or down arrow and the $\vdash$. The $\uparrow$-phase, also called the asynchronous phase, the part of a proof where invertible introduction rules are applied: when reading inference rules from conclusion to premises, invertible introduction rules are applied in any order and to exhaustion. If a positive formula appears to the left of the $\uparrow$, then that formula is delayed (stored) on the right of the $\uparrow$. Once all negative formula has been processed and all positive formulas have been stored, the $\downarrow$-phase, also called the asynchronous phase, is started. In particular, the decide rule selects one positive formula from storage and focuses on performing introduction rules on it and its positive subformulas. The following theorem is proved by Baelde in [3].

Theorem 12 (Soundness and Completeness of $\mu$MALLF). Let $Q$ be a $\mu$MALL formula. $\vdash Q$ is provable in $\mu$MALL if and only if $\vdash Q \uparrow \cdot$ is provable in $\mu$MALLF.

The focused proof system $\mu$LKF in Figure 3 is the result of modifying the proof system $\mu$MALLF to build the weakening rule into the $\top$, $\mathcal{=}$, and init rules (that is, by allowing the storage to be non-empty) and to build in the contraction rule into the $\otimes$ and decide rules. While $\mu$LKF is clearly a sound proof system for classical arithmetic, it is not known whether or not it is complete. As Girard points out in [8], the completeness of such a focused (cut-free) proof system would allow the extraction of the constructive context of a classical $\Pi^0_2$ theorems and we should not expect such a result to follow from the usual ways that we prove cut-elimination and the complete of focusing.

We note that all the metatheory in this paper regarding $\mu$LK would have been simplified if, in fact, $\mu$LKF is complete for $\mu$LK. In this sense, reasoning about proofs in $\mu$MALL is much easier since we can make use of the structure provided by $\mu$LALL.

10 Conclusion

We have used $\mu$MALL as linearized approach to arithmetic. We have shown that various computational logic topics—generalized unification problems, Horn clause provability, model checking queries, computing general recursive functions—can be captured correctly within this linearized arithmetic. Thus, these activities can be considered as relying on linear and not classical proof theory principles.
A linear logic approach to arithmetic

Asynchronous introduction rules (involves only \(\uparrow\)-sequents)

\[
\begin{align*}
\Gamma & \vdash \Delta \uparrow \Gamma \\
\vdash \bot, \Delta \uparrow \Gamma & \quad \vdash \top, \Delta \uparrow \Gamma \\
\vdash P, Q, \Delta \uparrow \Gamma & \quad \vdash \top, \Delta \uparrow \Gamma \\
\vdash P, \Delta \uparrow \Gamma & \quad \vdash Q, \Delta \uparrow \Gamma & \quad \vdash P, \Delta \uparrow \Gamma & \quad \vdash Q, \Delta \uparrow \Gamma & \quad \vdash P \& Q, \Delta \uparrow \Gamma
\end{align*}
\]

\[
\vdash s \neq t, \Delta \uparrow \Gamma \\
\vdash s \neq t, \Delta \uparrow \Gamma \\
\vdash S\vec{t}, \Delta \uparrow \Gamma & \quad \vdash BS\vec{x}, \Delta \uparrow \Gamma & \quad \vdash \forall x. P(x), \Delta \uparrow \Gamma & \quad \vdash \forall x. P(x), \Delta \uparrow \Gamma
\]

\[
\frac{\vdash \Delta \uparrow \Gamma \quad \theta = \text{mgw}(s, t)}{\vdash \Delta \uparrow \Gamma}
\]

Synchronous introduction rules (involves only \(\perp\)-sequents)

\[
\begin{align*}
\vdash 1 \perp & \quad \vdash P \perp \Gamma \quad \vdash Q \perp \Gamma' & \quad \vdash P \perp \Gamma \quad \vdash P \perp \Gamma
\end{align*}
\]

\[
\vdash t = i \perp & \quad \vdash \exists x. P(x) \perp \Gamma & \quad \vdash \forall x. P(x) \perp \Gamma & \quad \vdash \forall x. P(x) \perp \Gamma
\]

\[
\frac{\vdash \mu B \vec{\xi}}{\mu B \vec{\xi}} \quad \frac{\vdash \mu B \vec{\xi}}{\mu B \vec{\xi}} \quad \frac{\vdash \mu B \vec{\xi}}{\mu B \vec{\xi}}
\]

Non-introduction rules

\[
\begin{align*}
\vdash \Delta \uparrow \Gamma, P & \quad \vdash P \perp \Gamma & \quad \vdash \mu \Gamma, F & \quad \vdash N \uparrow \Gamma
\end{align*}
\]

\textbf{Figure 2} \(\mu\text{MALLF}.\) In the last three rules, \(P\) is a positive formula and \(N\) is a negative formula.

We have shown also that the notion of polarity, that first arose in linear logic, has an interesting connection with the familiar notion of arithmetical hierarchy. Such a refinement to the proof theory of Peano arithmetic should be a significant aid when one turns to implementing proof search in arithmetic.

References

ASYNCHRONOUS INTRODUCTION RULES (INVOLVE ONLY $\parallel$-SEQUENTS)

\[
\begin{align*}
\vdash & \Delta \parallel \Gamma & \vdash & P, Q, \Delta \parallel \Gamma & \vdash & P, \Delta \parallel \Gamma & \vdash & Q, \Delta \parallel \Gamma \\
\vdash & \bot, \Delta \parallel \Gamma & \vdash & P \otimes Q, \Delta \parallel \Gamma & \vdash & \top, \Delta \parallel \Gamma & \vdash & P \& Q, \Delta \parallel \Gamma
\end{align*}
\]

\[
\frac{\vdash \Delta \parallel \Gamma}{\vdash \Delta \theta \parallel \Gamma} \quad \frac{\vdash P, \Delta \parallel \Gamma}{\vdash \forall x. P, \Delta \parallel \Gamma}
\]

\[
\begin{align*}
\vdash & \exists \overline{x}. P \parallel \Gamma & \vdash & \forall \overline{x}. P \parallel \Gamma
\end{align*}
\]

\[
\frac{\vdash S \parallel \Delta \parallel \Gamma}{\vdash \overline{B} \overline{x}, (S \overline{x}) \parallel : \text{induct}} \quad \frac{\vdash \Delta \parallel \Gamma}{\vdash \forall \overline{B}, \overline{x} \parallel : \text{freeze}}
\]

SYNCHRONOUS INTRODUCTION RULES (INVOLVE ONLY $\gg$-SEQUENTS)

\[
\begin{align*}
\vdash & P \gg \Gamma & \vdash & Q \gg \Gamma & \vdash & P_0 \parallel P_1 \gg \Gamma & \vdash & P_1 \gg \Gamma & \vdash & P_0 \gg \Gamma \\
\vdash & \top \gg \Gamma & \vdash & \exists x. P x \gg \Gamma & \vdash & \mu B \overline{x} \gg \Gamma & \vdash & \nu B \overline{x} \gg \Gamma & \vdash & \mu B \overline{x} \gg \nu B \overline{x}, \Gamma
\end{align*}
\]

NON-INTRODUCTION RULES

\[
\begin{align*}
\vdash & \parallel \Gamma, P \quad \text{store} & \vdash & P \parallel \Gamma, P \quad \text{decide} & \vdash & N \parallel \Gamma \quad \text{release}
\end{align*}
\]

\[\blacksquare\] Figure 3 μLKF. In the last three rules, $P$ is a positive formula and $N$ is a negative formula.

A linear logic approach to arithmetic

