Kalpa Publications in Computing
Volume 18, 2024, Pages 117–130
LPAR 2024 Complementary Volume

On Proof Schemata and Primitive Recursive Arithmetic

Alexander Leitsch[1], Anela Lolić[2]*, and Stella Mahler[3]†

[1] Institute of Logic and Computation,
TU Wien, Vienna, Austria
leitsch@logic.at
[2] Kurt Gödel Society,
Institute of Logic and Computation,
TU Wien, Vienna, Austria
anela@logic.at
[3] Institute of Logic and Computation,
TU Wien, Vienna, Austria
stella@logic.at

Abstract

Inductive proofs can be represented as a proof schemata, i.e. as a parameterized sequence of proofs defined in a primitive recursive way. Applications of proof schemata can be found in the area of automated proof analysis where the schemata admit (schematic) cut-elimination and the construction of Herbrand systems. This work focuses on the expressivity of proof schemata as defined in [10]. We show that proof schemata can simulate primitive recursive arithmetic as defined in [12]. Future research will focus on an extension of the simulation to primitive recursive arithmetic using quantification as defined in [7]. The translation of proofs in arithmetic to proof schemata can be considered as a crucial step in the analysis of inductive proofs.

1 Introduction

Most interesting mathematical proofs contain applications of mathematical induction, making inductive theorem proving and the analysis of inductive proofs a crucial topic in computational logic. As Herbrand's theorem fails in presence of induction, an automated analysis of inductive proofs automated requires the use of novel frameworks and techniques. In [1] the method CERES (cut-elimination by resolution) was applied to analyze Fürstenberg's proof of the infinitude of primes. For this anaysis the original CERES method for first-order logic defined in [2] was extended to proof schemata (recursive representations of infinite sequences of proofs) ; the proof schema representing Fürstenberg's proof, still not fully formalized but represented on the mathematical meta-level, was subjected to cut-elimination via CERES resulting in the construction of (what was later called) a Herbrand system (an infinite sequence of Herbrand instances

*Partially supported by FWF project I-5848-N.
†Partially supported by FWF project I-5848-N.

represented by primitive recursion); this Herbrand system represented Euclid's construction of primes.

A first thorough analysis of an inductive (schematic) CERES method can be found in [9]; the inductive proofs investigated in this paper are those representable by a single parameter - in the formalisation by a *proof schema*. Here also the first concept of a *Herbrand system* was developed; it is essentially an extension of Herbrand's theorem from single proofs to a (recursively defined) infinite sequence of proofs. This definition of a Herbrand system represented the first step to extend Herbrand's theorem to inductive proofs. In [10] the approach in [9] was extended to arbitrary many induction parameters thus considerably increasing the strength of the method.

The schematic CERES method is capable of performing cut-elimination in presence of induction. There are other approaches to inductive inference where cut-elimination is possible as well. We just mention the works of Brotherston and Simpson [3] [4] and of McDowell and Miller [11]. However, though these approaches present inductive calculi with corresponding cut-elimination methods they do not allow the construction of any Herbrand structures (in particular of Herbrand systems) as described in [9] and [10].

In this paper we analyze the expressivity of proof schemata with arbitrary many parameters as defined in [10]. We prove that these schemata simulate primitive recursive arithmetic as defined in [12], a quantifier-free arithmetic using an induction rule. That means, via the translation, all arithmetical proofs formalizable in this arithmetic are candidates for an analyis via the schematic CERES method. For future work we plan to consider an extension of this arithmetic to the primitive recursive arithmetic as defined in [7], where the induction formulas are still quantifier-free but otherwise the introduction of quantifiers is admitted.

2 Schematic Language

The proof schemata we are considering are based on a many-sorted version of classical first-order logic and definitions via primitive recursion as introduced in [10]. Due to space limitations, we refer the interested reader to [10] and to [8] for formal definitions and details, and will present here only the most crucial notions and examples.

The first sort we consider is ω, in which every ground term normalizes to a *numeral*, i.e. a term inductively constructable over the signature $\Sigma_\omega = \{0, s(\cdot)\}$ as $N \Rightarrow s(N) \mid 0$, s.t. $s(N) \neq 0$ and $s(N) = s(N') \rightarrow N = N'$. Natural numbers ($\mathbb{N}$) will be denoted by lower-case Greek letters (α, β, γ, etc), the numeral $s^\alpha 0$, $\alpha \in \mathbb{N}$, will be written as $\bar{\alpha}$. The set of numerals is denoted by *Num*.

The ω sort includes a countable set of variables $\mathcal{N}$, called *parameters*. Parameters are denoted by $k, l, n, m, k_1, k_2, \ldots, l_1, l_2, \ldots, n_1, n_2, \ldots, m_1, m_2, \ldots$. The set of parameters occurring in an expression E is denoted by $\mathcal{N}(E)$. The set of *free ω-terms*, denoted by $\mathcal{T}_0^\omega$ contains all terms inductively constructable over Σ_ω and $\mathcal{N}$ as:

- If $t \in \mathcal{N}$ or $t \in Num$, then $t \in \mathcal{T}_0^\omega$.
- If $t \in \mathcal{T}_0^\omega$, then $s(t) \in \mathcal{T}_0^\omega$.

Moreover, the ω sort allows *defined function symbols*, the set of which will be denoted by $\hat{\Sigma}_\omega$. These symbols will be denoted using $\hat{\cdot}$ and have a fixed finite arity. The set of *ω-terms*, denoted by T^ω contains all terms inductively constructable over Σ_ω, $\hat{\Sigma}_\omega$, and $\mathcal{N}$, i.e.

- If $t \in T_0^\omega$, then $t \in T^\omega$.

- If $t_1, \cdots t_\alpha \in T^\omega$ and $\hat{f} \in \hat{\Sigma}_\omega$, s.t. $\hat{f}$ has arity $\alpha \geq 1$, then $\hat{f}(\overrightarrow{t}_\alpha) \in T^\omega$.

To every defined function symbol $\hat{f} \in \hat{\Sigma}_\omega$ of arity $\alpha + 1$ there exists a set of two defining equations of the form $D(\hat{f}) =$

$$\{\hat{f}(n_1, \ldots, n_\alpha, \bar{0}) = \hat{f}_B,\ \hat{f}(n_1, \ldots, n_\alpha, s(m+1)) = \hat{f}_S\{\xi \leftarrow \hat{f}(n_1, \ldots, n_\alpha, m)\}$$

where $\mathcal{N}(\hat{f}_B) \subseteq \{n_1, \ldots, n_\alpha\}$, $\mathcal{N}(\hat{f}_S) \subseteq \{n_1, \ldots, n_\alpha, \xi\}$ and $\hat{f}_B, \hat{f}_S$ contain only defined function symbols which are smaller than $\hat{f}$ (for a precise definition of the ordering see [5]).

Example 1. *For $\widehat{p} \in \Sigma_\omega$, $D(\widehat{p}) = \{\widehat{p}(\bar{0}) = \bar{0},\ \widehat{p}(s(m)) = m\}$, $\hat{p}_B = \bar{0}$, $\hat{p}_S = m$.*
Let $\hat{f}, \hat{g} \in \Sigma_\omega$ s.t. $\hat{f}$ is smaller than $\hat{g}$. We define $D(\hat{f})$ as

$$\hat{f}(n, \bar{0}) = \hat{f}_B, \quad \hat{f}(n, s(m)) = \hat{f}_S\{\xi \leftarrow \hat{f}(n, m)\}$$

for $\hat{f}_B = n$ and $\hat{f}_S = s(\xi)$. Then, obviously, $\hat{f}$ defines $+$. Now we define $D(\hat{g})$ as

$$\hat{g}(n, \bar{0}) = \hat{g}_B, \quad \hat{g}(n, s(m)) = \hat{g}_S\{\xi \leftarrow \hat{g}(n, m)\}$$

where $\hat{g}_B = \bar{0}$ and $\hat{g}_S = \hat{f}(n, \xi)$. Then $\hat{g}$ defines $*$. In both cases ξ is any fresh parameter in $\mathcal{N}$. We say that the corresponding theory *is $\Big(\{\hat{p}, \hat{f}, \hat{g}\}, \{\hat{g}\}, D(\hat{p}) \cup D(\hat{f}) \cup D(\hat{g})\Big)$ (for a formal definition see [8], page 5, Definition 2.12).*

The second sort, the ι-sort for individuals, also has two associated signatures, the set of free function symbols, Σ_ι, and the set of *defined function symbols*, $\hat{\Sigma}_\iota$. Besides parameters we also have the sets of individual variables V. The set of ι-terms, denoted by $\mathcal{T}^\iota$ is inductively constructed from Σ_ι, $\hat{\Sigma}_\iota$, and V as:

- $V \subseteq \mathcal{T}^\iota$.
- If $f \in \Sigma_\iota$, f is α-place and $t_1, \ldots, t_\alpha \in \mathcal{T}^\iota$ then $f(t_1, \ldots, t_\alpha) \in \mathcal{T}^\iota$.
- If $s_1, \cdots, s_\alpha \in \mathcal{T}^\iota$, $t_1, \cdots, t_{\beta+1} \in T^\omega$, $\hat{f} \in \hat{\Sigma}_\iota$, s.t. $\hat{f}$ has arity $\alpha + \beta + 1$ for $\alpha, \beta \geq 0$, then $\hat{f}(s_1, \cdots, s_\alpha, t_1, \cdots, t_{\beta+1}) \in \mathcal{T}^\iota$.

Like for T^ω there is a set of two defining equations for every symbol $\hat{f} \in \mathcal{T}^\omega$; for details we refer to [5] and [8]. As an example consider

Example 2. *Let $f \in \Sigma_\iota$, $\hat{f} \in \hat{\Sigma}_\iota$ and $x \in V$. We define $D(\hat{f})$ as*

$$\hat{f}(x, \bar{0}) = x, \quad \hat{f}(x, m+1) = f(\hat{f}(X, m)).$$

Considering $\hat{f}_B, \hat{f}_S$ like for T^ω, we get $\hat{f}_B = x, \hat{f}_S = f(\xi)$. E.g. $\hat{f}(x, \bar{3})$ rewrites to the term $f(f(f(x)))$.

The third and final sort we consider is that of *formulas* which will be denoted by o. Formulas are constructed using the signature $\Sigma_o = \{\neg, \wedge, \vee\}$, a countably infinite set of predicate symbols $\mathcal{P}$ with fixed and finite arity, and a countably infinite set of formula variables V^F. The set of formulas, denoted by $\mathcal{T}_V^o$ is constructed inductively as:

- If $t \in V^F$, then $t \in \mathcal{T}_V^o$.

- If $t_1, \ldots, t_\alpha \in T^\iota$ and $P \in \mathcal{P}$ s.t. P has arity $\alpha \geq 0$, then $P(t_1, \ldots, t_\alpha) \in \mathcal{T}_V^o$.
- If $t \in \mathcal{T}_V^o$, then $\neg t \in \mathcal{T}_V^o$.
- If $t_1, t_2 \in \mathcal{T}_V^o$ and $\star \in \{\vee, \wedge\}$, then $t_1 \star t_2 \in \mathcal{T}_V^o$.

The set of formulas in $\mathcal{T}_V^o$ which do not contain formula variables is denoted by $\mathcal{T}_0^o$.

Formula schemata are constructed using formula terms by allowing *defined predicate symbols* to occur. Similarly as in the previous cases, defined symbols will be denoted by $\hat{\cdot}$ and have a fixed finite arity. The set of defined predicate symbols is denoted by $\hat{\mathcal{P}}$. The set of formula schemata is denoted by $\mathcal{T}_o(\Sigma_o, \mathcal{P}, V^F, V^G, \mathcal{N}, \hat{\mathcal{P}})$ and is constructed inductively as:

- If $t \in \mathcal{T}_V^o$, then $t \in \mathcal{T}^o$.
- If $t_1, \cdots, t_\alpha \in \mathcal{T}^\iota$, $\hat{p} \in \hat{\mathcal{P}}$, $s_1, \ldots, s_{\beta+1} \in T^\omega$ s.t. $\hat{p}$ has arity $\alpha + \beta + 1$ for $\alpha, \beta \geq 0$, then $\hat{p}(t_1, \ldots, t_\alpha, s_1, \ldots, s_{\beta+1}) \in \mathcal{T}^o$.
- If $t \in \mathcal{T}^o$, then $\neg t \in \mathcal{T}^o$.
- If $t_1, t_2 \in \mathcal{T}^o$ and $\star \in \{\vee, \wedge\}$, then $t_1 \star t_2 \in \mathcal{T}^o$.

For every defined symbol $\hat{p} \in \hat{\Sigma}_o$ there exists a set of defining equations $D(\hat{p})$ which expresses a primitive recursive definition of $\hat{p}$.

Definition 1 (defining equations)**.** *Let $\hat{p} \in \hat{\sigma}_o$. We define a set $D(\hat{p})$ consisting of two equations:*

$$\begin{aligned} \hat{p}(x_1, \ldots, x_\alpha, n_1, \ldots, n_\beta, \bar{0}) &= \hat{p}_B, \\ \hat{p}(x_1, \ldots, x_\alpha, n_1, \ldots, n_\beta, s(m)) &= \hat{p}_S\{\xi \leftarrow \hat{p}(x_1, \ldots, x_\alpha, n_1, \ldots, n_\beta, m)\}, \text{ where} \end{aligned}$$

1) If $\hat{p}$ is minimal (there is no smaller $\hat{q} \in \hat{\Sigma}_o$):

a) $\hat{p}_B \in \mathcal{T}_0^o$, $\hat{p}_S \in \mathcal{T}_V^o$.

b) $|V^F(\hat{p}_S)| \leq 1$.

2) If $\hat{f}$ is non-minimal: $\hat{p}_B, \hat{p}_S \in \mathcal{T}^o$ where $\hat{p}_B, \hat{p}_S$ may contain only defined predicate symbols smaller than $\hat{p}$. Moreover, $|V^F(\hat{p}_S)| \leq 1$ and $|V^F(\hat{p}_B)| = 0$.

Additionally, $\mathcal{N}(\hat{p}_B) \subseteq \{n_1, \ldots, n_\beta\}$, $\mathcal{N}(\hat{p}_S) \subseteq \{n_1, \ldots, n_\beta\} \cup \{m, \xi\}$ and the only individual variables occurring in $\hat{p}_B$ and $\hat{p}_S$ are in $\{x_1, \ldots, x_\alpha\}$. We define $D^o = \bigcup\{D(\hat{p}) \mid \hat{p} \in \hat{\Sigma_o}\}$.

It is easy to see that, given any parameter assignment, all terms in T^ω evaluate to numerals. The defined symbols in our language introduce an equational theory and without restrictions on the use of these equalities the word problem is undecidable. In the definitions above the equations can be oriented to terminating and confluent rewrite systems and thus termination of the evaluation procedure is easily verified [5].

Definition 2 (parameter assignment)**.** *A function $\sigma \colon \mathcal{N} \to Num$ is called a* parameter assignment. *σ is extended to $\mathcal{T}^\omega$ homomorphically:*

- *$\sigma(\bar{\beta})\!\downarrow = \bar{\beta}$ for numerals $\bar{\beta}$.*
- *$\sigma(s(t))\!\downarrow = s(\sigma(t)\!\downarrow)$*

- $\sigma(\hat{f}(\overrightarrow{t_\alpha}))\downarrow = \hat{f}(\sigma(\overrightarrow{t_\alpha})\downarrow)\downarrow$ *for* $\hat{f} \in \Sigma_\omega$ *and* $\overrightarrow{t_\alpha} \in T^\omega$.

The set of all parameter assignments is denoted by $\mathcal{S}$.

Note that parameter assignments can be extended to ι and o terms in an obvious way. While numeric terms evaluate to numerals under parameter assignments, terms in T^ι evaluate to terms in T_0^ι, i.e. to ordinary first-order terms, and terms in T^o evaluate to terms in T_0^o, i.e. Boolean expressions. Evaluations are denoted by$\downarrow$, e.g. for $F \in \mathcal{T}^o$ $\sigma(F)\downarrow$ is a formula in $\mathcal{T}_0^o$.

3 Proof Schema

The general idea of a proof schema is to represent a proof by a finite description of an infinite sequence of proofs. Assume a proof φ of the end-sequent $\vdash \forall x A(x)$ that uses an induction inference. Instead of considering the proof φ, we instead consider the infinite sequence of proofs $\varphi_0,\ \varphi_1,\ \varphi_2,\ \ldots$ of end-sequents $\vdash\ A(0),\ \vdash\ A(1),\ \vdash\ A(2)\ \ldots.$ The task is to find a finite description of this infinite sequence of proofs, the proof schema. A proof schema always represents a parameterized sequence, and an evaluation under a parameter assignment n results in the proof φ_n of $\vdash A(n)$. The underlying problem, that initially lead to the development of proof schemata, is to be able to analyze inductive proofs. Indeed, each of the proofs φ_0, φ_1, φ_2, ... is a simple **LK**-proof without induction inferences, and thus enjoys cut-elimination resulting in an analytic proof. The concept of proof schema was initially introduced in [6, 9] to address schemata involving a single parameter. Subsequently, in [10], it was expanded to accommodate an arbitrary number of parameters.

Formally, proof schemata are constructed using proofs in an extension of **LK** by an equational theory. First, let us define the concept of schematic sequents.

Definition 3 (schematic sequents)**.** *A schematic sequent is a sequent of the form* $F_1, \ldots, F_\alpha \vdash G_1, \ldots, G_\beta$ *where the* F_i *and* G_j *for* $1 \leq i \leq \alpha$ *and* $1 \leq j \leq \beta$ *are formula schemata. Let* $S\colon F_1, \ldots, F_\alpha \vdash G_1, \ldots, G_\beta$ *be a schematic sequent and* σ *a parameter assignment. Then the evaluation of* S *under* σ *is* $\sigma(S)\downarrow\colon \sigma(F_1)\downarrow, \ldots, \sigma(F_\alpha)\downarrow \vdash \sigma(G_1)\downarrow, \ldots, \sigma(G_\beta)\downarrow$.

In this work we restrict the end-sequent schemata to skolemized sequents in prenex form.

Definition 4. *Let* $\mathcal{E}$ *be an equational theory. We extend the calculus* **LK** *by the* $\mathcal{E}$ *inference rule* $\dfrac{S(t)}{S(t')}\mathcal{E}$ *where the term or input term schema* t *in the schematic sequent* S *is replaced by a term or input term schema* t' *for* $t = t' \in \mathcal{E}$ *(or* $t \leftrightarrow t' \in \mathcal{E}$*).*

The definitions below will use the schematic standard axiom set $\mathcal{A}_s$.

Definition 5 (schematic standard axiom set)**.** *Let* $\mathcal{A}_s$ *be the smallest set of schematic sequents that is closed under substitution containing all sequents of the form* $A \vdash A$ *for arbitrary atomic formula schemata* A*. Then* $\mathcal{A}_s$ *is called the schematic standard axiom set.*

Schematic derivations can be understood as parameterized sequences of **LK**-derivations where new kinds of axioms (in the form of labeled sequents) are included. These labeled sequents serve the purpose to establish recursive call structures in the proof. For constructing schematic derivations we introduce a countably infinite set Δ of *proof symbols* which are used to label the individual proofs of a proof schema. A particular proof schema uses a finite set of proof symbols $\Delta^* \subset \Delta$. We assign an arity $A(\delta)$ to every $\delta \in \Delta^*$, $A(\delta)$ is the arity of the input parameters for the proof labeled by δ. Also, we need a concept of proof labels which serve the purpose to relate some leafs of the proof tree to recursive calls.

Definition 6 (proof label)**.** *Let $\delta \in \Delta$ and ϑ be a parameter substitution. Then the pair (δ, ϑ) is called a proof label.*

Definition 7 (labeled sequents and derivations)**.** *Let S be a schematic sequent and (δ, ϑ) a proof label, then $(\delta, \vartheta)\colon S$ is a* labeled sequent*. A* labeled *derivation is a derivation π where all leaves are labeled.*

In the definition below we will define a proof schema over a base-case proof (for parameter 0) and a step-case proof (for parameter $m+1$), where initial sequents are either axioms, or end-sequents from previously defined (base- or step-case) proofs. In general, the step-case proof for some proof symbol δ uses as initial sequent its own end-sequent, but under a parameter assignment m. Evaluating a schematic derivation means that initial sequents, which are no axioms, have to be replaced by their derivations.

Definition 8 (parameter replacement)**.** *Let $\vec{m}, \vec{n}$ be tuples of parameters. A parameter replacement on $\vec{n}$ with respect to $\vec{m}$ is a replacement substituting every parameter p in $\vec{n}$ by a term t_p, where the parameters of $t_p \in T^\omega$ are in $\vec{m}$.*

Definition 9 (schematic deduction and proof schema)**.** *Let $\mathcal{D}$ be the tuple $(\delta_0, \Delta^*, \Pi)$. $\mathcal{D}$ is called a schematic deduction from a finite set of schematic sequents $\mathcal{S}$ if the following conditions are fulfilled:*

- *Δ^* is a finite subset of Δ.*
- *$\delta_0 \in \Delta^*$, and $\delta_0 > \delta'$ for all $\delta' \in \Delta^*$ such that $\delta' \neq \delta_0$. δ_0 is called the main symbol.*
- *To every $\delta \in \Delta^*$ we assign a parameter tuple $\vec{n}_\delta$ of pairwise different parameters (called the passive parameters), and a parameter m_δ (called the active parameter).*
- *Π is a set of pairs $\{(\Pi(\delta, \vec{n}_\delta, m_\delta), S(\delta, \vec{n}_\delta, m_\delta)\}$, where $S(\delta, \vec{n}_\delta, m_\delta)$ is a schematic sequent, and*
$$\Pi(\delta, \vec{n}_\delta, m_\delta) = \{(\delta, \vec{n}_\delta, m_\delta) \rightarrow \rho(\delta, \vec{n}_\delta, m_\delta)\},$$
where $\rho(\delta, \vec{n}_\delta, 0) = \rho_0(\delta, \vec{n}_\delta)$, and $\rho(\delta, \vec{n}_\delta, s(m_\delta)) = \rho_1(\delta, \vec{n}_\delta, m_\delta)$, and there exists a (possibly empty) finite set of schematic sequents $\mathcal{C}(\delta)$ such that
  1. *$\rho_0(\delta, \vec{n}_\delta)$ is a deduction of $S(\delta, \vec{n}_\delta, 0)$ from $\mathcal{S} \cup \mathcal{C}(\delta)$,*
  2. *$\rho_1(\delta, \vec{n}_\delta, m_\delta)$ is a deduction of $S(\delta, \vec{n}_\delta, m_\delta + 1)$ from $\{(\delta, \Psi)\colon S(\delta, \vec{n}_\delta, m_\delta)\} \cup \mathcal{S} \cup \mathcal{C}(\delta)$, where (δ, Ψ) is a label, and Ψ the empty parameter replacement,*
  3. *for all $S' \in \mathcal{C}(\delta)$, $S' = (\delta', \Psi)\colon S(\delta', \vec{n}_{\delta'}, m_{\delta'})\Psi$ where (δ', Ψ) is a label, $\delta' \in \mathbf{\Delta}^*$ with $\delta > \delta'$ and Ψ is a parameter replacement on $(\vec{n}_{\delta'}, m_{\delta'})$ w.r.t. $(\vec{n}_\delta, m_\delta)$ such that the conditions 1. and 2. hold for δ' .*

If $\mathcal{S} = \mathcal{A}_s$ we call $\mathcal{D}$ a proof schema *of $S(\delta_0, \vec{n}_{\delta_0}, m_{\delta_0})$.*

As our formalism is capable of handling several induction parameters, we can easily formalize common proofs in Peano arithmetic, as for instance commutativity, as a proof schema. In the following examples we will use the standard Peano axioms.

Definition 10 (Peano axioms)**.** *The Peano axioms are defined as*

$$\begin{aligned} A1 &: \vdash 0 \neq s(x) \\ A2 &: \vdash s(x) = s(y) \rightarrow x = y \\ A3 &: \vdash x + 0 = x \\ A4 &: \vdash x + s(y) = s(x + y) \\ A5 &: \vdash x \times 0 = 0 \\ A6 &: \vdash x \times s(y) = x \times y + x \end{aligned}$$

Frequently, we will denote $s(0)$ *as* 1 *and add the axiom*

$$A7 \quad : \quad \vdash s(x) = x + 1$$

which can be proven easily using $A3$*,* $A4$ *and the definition of* 1*.*

Example 3. *In the following examples we will use the schematic standard axiom set extended by Peano axioms and the usual equality rules (denoted by* $\mathcal{E}$*).It is easy to see that the axioms, along with valid equality rules, translate into valid inference rules in our calculus. This translation is linear in the number of inferences. Therefore, we refrain from providing a detailed translation.*

Proof of 0 is a left-identity*: We define a proof schema of* $\vdash 0 + m = m$*. Let* $\mathcal{D} = \{(\delta, \rho(\delta, 0), \rho(\delta, m+1))\}$ *where* $S(\delta) = \vdash 0 + m = m$ *and we define* $\rho(\delta, 0)$ *as follows:*

$$\dfrac{\vdash 0 = 0}{\vdash 0 + 0 = 0}\, A3$$

$\rho(\delta, m+1)$ *is defined as follows:*

$$\dfrac{\dfrac{(\delta, \emptyset) \colon S(\delta)}{\vdash s(0 + m) = s(m)}}{0 + s(m) = s(m)}\, A4$$

Proof of associativity*: We define a proof schema of* $\vdash (a + b) + m = a + (b + m)$*. Let* $\mathcal{D}_1 = \{(\delta_1, \rho(\delta_1, a, b, 0), \rho(\delta_1, a, b, m+1))\}$ *where* $S(\delta_1) = \vdash (a + b) + m = a + (b + m)$ *and we define* $\rho(\delta_1, a, b, 0)$ *as follows:*

$$\dfrac{\dfrac{\vdash a + b = a + b}{\vdash a + b = a + (b + 0)}\, A3}{\vdash (a + b) + 0 = a + (b + 0)}\, A3$$

$\rho(\delta_1, a, b, m+1)$ *is defined as follows:*

$$\dfrac{\dfrac{\dfrac{\dfrac{(\delta_1, \emptyset) \colon S(\delta_1)}{\vdash s((a + b) + m) = s(a + (b + m))}}{\vdash s((a + b) + m) = a + s(b + m)}\, A4}{\vdash s((a + b) + m) = a + (b + s(m))}\, A4}{\vdash (a + b) + s(m) = a + (b + s(m))}\, A4$$

To prove commutativity, we first need to define a proof schema of $\vdash m + 1 = 1 + m$*. Let* $\mathcal{D}_2 = \{(\delta_2, \rho(\delta_2, 0), \rho(\delta_2, m+1))\} \cup \mathcal{D}$*, where* $\delta_2 > \delta$*,* $S(\delta_2) = \vdash m + 1 = 1 + m$ *and we define* $\rho(\delta_2, 0)$ *as follows:*

$$\dfrac{(\delta,\{m\leftarrow 1\})\colon S(\delta)\{m\leftarrow 1\}}{\vdash 0+1=1+0}\,A3$$

$\rho(\delta_2, m+1)$ is defined as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\delta_2,\emptyset)\colon S(\delta_2)}{\vdash s(m+1)=s(1+m)}\qquad \dfrac{\vdash 1+s(m)=1+s(m)}{\vdash s(1+m)=1+s(m)}\,A4}{\vdash s(m+1)=s(0)+s(m)}\,\mathcal{E},def}{\vdash s((m+1)+0)=s(0)+s(m)}\,A3}{\vdash s(s(m)+0)=s(0)+s(m)}\,A7}{\vdash s(m)+s(0)=s(0)+s(m)}\,A4}{\vdash s(m)+1=1+s(m)}\,def$$

Now we define a proof schema of $\vdash n+m'=m'+n$. Let $\mathcal{D}_3=\{(\delta_3,\rho(\delta_3,n,0),\rho(\delta_3,n,m'+1))\}\cup\mathcal{D}_2\cup\mathcal{D}_1$, where $\delta_3>\delta_2$, $\delta_3>\delta_1$, $S(\delta_3)=\vdash n+m'=m'+n$ and we define $\rho(\delta_3,n,0)$ as follows:

$$\dfrac{\dfrac{(\delta,\{m\leftarrow n\})\colon S(\delta)\{m\leftarrow n\}}{\vdash n=0+n}\,\mathcal{E}}{\vdash n+0=0+n}\,A3$$

$\rho(\delta_3, n, m'+1)$ is defined as follows:

$$\dfrac{\begin{array}{c}(\delta_1,\{a\leftarrow n,b\leftarrow m',m\leftarrow 1\})\colon\\ S(\delta_1)\{a\leftarrow n,b\leftarrow m',m\leftarrow 1\}\\ \vdash n+(m'+1)=(n+m')+1\end{array}\qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}S(\delta_3)\\ \vdash n+m'=m'+n\end{array}}{\vdash s(n+m')=s(m'+n)}\qquad \begin{array}{c}\phi_1\\ \vdash s(m'+n)=(m'+1)+n\end{array}}{\vdash s(n+m')=(m'+1)+n}\,\mathcal{E}}{\vdash s((n+m')+0)=(m'+1)+n}\,A3}{\vdash (n+m')+s(0)=(m'+1)+n}\,A4}{\vdash (n+m')+1=(m'+1)+n}\,def}{\vdash n+(m'+1)=(m'+1)+n}\,\mathcal{E}$$

where ϕ_1 is

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}(\delta_1,\{a\leftarrow m',b\leftarrow 1,m\leftarrow n\})\colon\\ S(\delta_1)\{a\leftarrow m',b\leftarrow 1,m\leftarrow n\}\\ \vdash m'+(1+n)=(m'+1)+n\end{array}\qquad \dfrac{\begin{array}{c}(\delta_2,\{m\leftarrow n\})\colon S(\delta_2)\{m\leftarrow n\}\\ \vdash n+1=1+n\end{array}}{\vdash 1+n=n+1}\,\mathcal{E}}{\vdash m'+(n+1)=(m'+1)+n}\,\mathcal{E}}{\vdash m'+s(n)=(m'+1)+n}\,def}{\vdash s(m'+n)=(m'+1)+n}\,A4$$

We are going to evaluate proof schemata under parameter assignments.

Definition 11 (evaluation of proof schema)**.** *Let $\mathcal{D}=(\delta_0,\Delta^*,\Pi)$ be a proof schema, and σ a parameter assignment. In defining the evaluation of the proof schema, $\mathcal{D}\sigma\downarrow$, we proceed by double induction on the ordering of proof symbols and the assignments σ.*

- *Let δ_i be a minimal element in Δ^*.*

1. $\sigma(m_{\delta_i}) = 0$.
   Then, by definition of a proof schema, $\rho_0(\delta_i, \vec{n}_{\delta_i})$ *is a proof with* **LK**-*inferences and inferences for defined symbols that contain schematic sequents. Let* $S_1, \dots S_n$ *be all the schematic sequents in* $\rho_0(\delta_i, \vec{n}_{\delta_i})$. *Then the evaluation of* $\rho_0(\delta_i, \vec{n}_{\delta_i})$ *under* σ *is denoted by* $\rho_0(\delta_i, \vec{n}_{\delta_i})\downarrow$ *and obtained by replacing all* $S_1, \dots, S_n$ *in* $\rho_0(\delta_i, \vec{n}_{\delta_i})$ *by* $\sigma(S_1)\downarrow, \dots, \sigma(S_n)\downarrow$ *and omitting the inferences for defined symbols.*
2. $\sigma(m_{\delta_i}) = \alpha > 0$.
   Evaluate all schematic sequents except the leaves $(\delta_i, \emptyset)\colon S(\delta_i, \vec{n}_{\delta_i}, m_{\delta_i})$ *under* σ. *Let* $\sigma[m_{\delta_i}/\alpha - 1]$ *be defined as* $\sigma[m_{\delta_i}/\alpha - 1](p) = \sigma(p)$ *for all* $p \neq m_{\delta_i}$ *and* $\sigma[m_{\delta_i}/\alpha - 1](m_{\delta_i}) = \alpha - 1$. *Then we replace the labeled sequent* $(\delta_i, \emptyset)\colon S(\delta_i, \vec{n}_{\delta_i}, m_{\delta_i})$ *by the proofs* $\rho_0(\delta_i, \vec{n}_{\delta_i})\sigma[m_{\delta_i}/\alpha - 1]\downarrow$ *if* $\alpha - 1 = 0$ *and by* $\rho_1(\delta_i, \vec{n}_{\delta_i}, m_{\delta_i})\sigma[m_{\delta_i}/\alpha - 1]\downarrow$ *if* $\alpha - 1 > 0$. *The result is an* **LK**-*proof.*

- $\delta_i \in \Delta^*$ *is not minimal.*

1. $\sigma(m_{\delta_i}) = 0$.
   Evaluate all schematic sequents except the labeled sequents of the form $(\delta', \Psi)\colon$ $S(\delta', \vec{n}_{\delta'}, m_{\delta'})\Psi$ *for* $\delta_i > \delta'$ *and the corresponding parameter replacement* Ψ *under* σ. *Then replace the labeled sequent* $(\delta', \Psi)\colon S(\delta', \vec{n}_{\delta'}, m_{\delta'})\Psi$ *by the proof* $\rho_0(\delta', \vec{n}_{\delta'})\Psi\sigma\downarrow$ *if* $\sigma(m_{\delta'}) = 0$ *and by the proof* $\rho_1(\delta', \vec{n}_{\delta'}, m_{\delta'})\Psi\sigma\downarrow$ *otherwise.*
2. $\sigma(m_{\delta_i}) = \alpha > 0$.
   As above, except for the labeled sequents $(\delta_i, \emptyset)\colon S(\delta_i, \vec{n}_{\delta_i}, m_{\delta_i})$ *which are replaced by the proof* $\rho_0(\delta_i, \vec{n}_{\delta_i})\sigma[m_{\delta_i}/\alpha - 1]$ *if* $\alpha - 1 = 0$ *and by the proof* $\rho_1(\delta_i, \vec{n}_{\delta_i}, m_{\delta_i})\sigma[m_{\delta_i}/\alpha - 1]$ *otherwise.*

$\mathcal{D}\sigma\downarrow$ *is defined as* $\rho_0(\delta_0, \vec{n}_{\delta_0})\sigma\downarrow$ *for the* $<$-*maximal symbol* δ_0 *if* $\sigma(m_{\delta_0}) = 0$, *and by* $\rho_1(\delta_0, \vec{n}_{\delta_0}, m_{\delta_0})\sigma\downarrow$ *if* $\sigma(m_{\delta_0}) > 0$.

4 Simulation of Primitive Recursive Arithmetic Through Proof Schemata

In [9], it was demonstrated that proof schemata are equivalent to a specific fragment of arithmetic known as *k-simple induction*. This variant restricts the introduction of new eigenvariables through induction. As we are dealing with proof schemata allowing an arbitrary number of parameters, we will now provide a simulation of primitive recursive arithmetic through proof schemata without parameter restriction. Following [12], we define the respective calculus as the propositional part of Gentzen's **LK** (see Figure 1) extended by an equational theory as in Definition 4 and the following induction rule:

$$\frac{\Gamma \vdash \Delta, F(0) \qquad \Gamma, F(y) \vdash \Delta, F(y+1)}{\Gamma \vdash \Delta, F(n)}\,(IND)$$

where y is a variable of sort ω, n is a variable of sort ω, and y does not occur in $\Gamma, \Delta, F(0)$.

Note that in [12], the induction variable is an arbitrary term. Our restriction to a variable of sort ω is equivalent, as it is easy to see that any arbitrary term can be simulated in the conclusion. Further, every sequent $S : \Gamma \vdash \Delta$ corresponds to an equivalent formula $\mathcal{F}(S) \coloneqq \bigvee \neg\Gamma \cup \Delta$. The calculus resulting from combining the rules from Figure 1, $\mathcal{E}$ and (IND) is denoted by **PRA**.

We will now show the translation from quantifier-free proof schemata to **PRA** and back.

$$\frac{\Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}\ (w:l) \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A}\ (w:r)$$

$$\frac{A, A, \Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}\ (c:l) \qquad \frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A}\ (c:r)$$

$$\frac{}{F \vdash F}\ (Axiom) \qquad \frac{\Gamma \vdash \Delta, F \qquad \Sigma, F \vdash \Pi}{\Gamma, \Sigma \vdash \Delta, \Pi}\ (cut)$$

$$\frac{\Gamma, F, G \vdash \Delta}{\Gamma, F \wedge G \vdash \Delta}\ (\wedge:l) \qquad \frac{\Gamma \vdash \Delta, F \qquad \Sigma \vdash \Pi, G}{\Gamma, \Sigma \vdash \Delta, \Pi, F \wedge G}\ (\wedge:r)$$

$$\frac{\Gamma, F \vdash \Delta \qquad \Sigma, G \vdash \Pi}{\Gamma, \Sigma, F \vee G \vdash \Delta, \Pi}\ (\vee:l) \qquad \frac{\Gamma \vdash \Delta, F, G}{\Gamma \vdash \Delta, F \vee G}\ (\vee:l)$$

$$\frac{\Gamma \vdash \Delta, F}{\Gamma, \neg F \vdash \Delta}\ (\neg:l) \qquad \frac{\Gamma, F \vdash \Delta}{\Gamma \vdash \Delta, \neg F}\ (\neg:r)$$

$$\frac{\Gamma \vdash \Delta, F \qquad \Sigma, G \vdash \Pi}{\Gamma, \Sigma, F \to G \vdash \Delta, \Pi}\ (\to:l) \qquad \frac{\Gamma, F \vdash \Delta, G}{\Gamma \vdash \Delta, F \to G}\ (\to:r)$$

Figure 1: The propositional part of Gentzen's **LK**

Lemma 1. *Let $\mathcal{D}$ be a proof schema with end-sequent S. Then there exists a **PRA** proof of S.*

Proof. Let $\mathcal{D} = \{(\delta_i, \rho(\delta_i, \vec{n}_i, 0), \rho(\delta_i, \vec{n}_i, m_i + 1)) \mid i \in \{1, ..., \alpha\}\}$ with $S(\delta_i) = S_i$ and if $i < j$ then $\delta_i > \delta_j$. Hence, $S(\delta_1) = S$.

We construct inductively **PRA** proofs of $\mathcal{F}(S_\gamma)$, starting with $\gamma = \alpha$. Assume we constructed **PRA** proofs $\xi_{\gamma+1}, ..., \xi_\alpha$ of $\mathcal{F}(S_{\gamma+1}), ..., \mathcal{F}(S_\alpha)$ respectively. Our aim is to construct a **PRA** proof of $\mathcal{F}(S_\gamma)$. In $\rho(\delta_\gamma, \vec{n}_\gamma, 0)$ replace any proof call of the form $(\delta_j, \Psi) : S(\delta_j)\Psi$ by $\xi_j\Psi$ to obtain proof ξ^B_γ. In $\rho(\delta_\gamma, \vec{n}_\gamma, m_{\gamma+1})$ replace any proof call of the form $(\delta_j, \Psi) : S(\delta_j)\Psi$ with $j \neq \gamma$ by $\xi_j\Psi$ and replace any self-referencing proof call of the form $(\delta_\gamma, \emptyset) : S(\delta_\gamma)$ by axiom $\mathcal{F}(S(\delta_\gamma)) \vdash \mathcal{F}(S(\delta_\gamma))$ to obtain proof ξ^S_γ. The desired proof is then constructed as follows:

$$\frac{\dfrac{\xi^B_\gamma}{\vdash \mathcal{F}(S_\gamma)\{m_\gamma \leftarrow 0\}} \qquad \dfrac{\xi^S_\gamma}{\mathcal{F}(S_\gamma)\{m_\gamma \leftarrow y\} \vdash \mathcal{F}(S_\gamma)\{m_\gamma \leftarrow y+1\}}}{\vdash \mathcal{F}(S_\gamma)}\ \text{(IND)}$$

Note that in case of a proof call which includes an instantiation, we use (cut) instead of (IND). Finally, we use cuts to derive S from the proof of $\mathcal{F}(S)$. □

Example 4. *To illustrate this, we provide a translation of the schematic proof of commutativity from Example 3 into **PRA**. The proof schema $\mathcal{D}_3$ is then translated into:*

$$\frac{\dfrac{\varphi_1}{\vdash n + 0 = 0 + n} \qquad \dfrac{\varphi_2}{n + z = z + n \vdash n + s(z) = s(z) + n}}{\vdash n + m' = m' + n}\ (IND)$$

where φ_1 is

$$
\dfrac{\dfrac{\dfrac{\dfrac{\vdash 0=0}{\vdash 0+0=0}\,A3 \qquad \dfrac{\dfrac{\dfrac{0+y=y\vdash 0+y=y}{0+y=y\vdash s(0+y)=s(y)}}{0+y=y\vdash 0+s(y)=s(y)}\,A4}{\vdash 0+n=n}\,(IND)}{\vdash n=0+n}\,\mathcal{E}}{\vdash n+0=0+n}\,A3
$$

and φ_2 *is*

$$
\dfrac{\dfrac{\dfrac{\psi_1}{\vdash n+(z+1)=(n+z)+1}}{\vdash n+s(z)=(n+z)+1}\,A7 \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{n+z=z+n\vdash n+z=z+n}{n+z=z+n\vdash s(n+z)=s(z+n)} \quad \dfrac{\psi_2}{\vdash s(z+n)=(z+1)+n}}{n+z=z+n\vdash s(n+z)=(z+1)+n}\,\mathcal{E}}{n+z=z+n\vdash s((n+z)+0)=(z+1)+n}\,A3}{n+z=z+n\vdash (n+z)+s(0)=(z+1)+n}\,A4}{n+z=z+n\vdash (n+z)+1=s(z)+n}\,def}{n+z=z+n\vdash n+s(z)=s(z)+n}\,\mathcal{E}
$$

with ψ_1

$$
\dfrac{\dfrac{\dfrac{\dfrac{\vdash n+z=n+z}{\vdash n+z=n+(z+0)}\,A3}{\vdash (n+z)+0=n+(z+0)}\,A3 \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(n+z)+0=n+(z+0)\vdash (n+z)+0=n+(z+0)}{(n+z)+0=n+(z+0)\vdash s((n+z)+0)=s(n+(z+0))}}{(n+z)+0=n+(z+0)\vdash s((n+z)+0)=n+s((z+0))}\,A4}{(n+z)+0=n+(z+0)\vdash s((n+z)+0)=n+(z+s(0))}\,A4}{(n+z)+0=n+(z+0)\vdash (n+z)+s(0)=n+(z+s(0))}\,A4}{(n+z)+0=n+(z+0)\vdash (n+z)+1=n+(z+1)}\,def}{\vdash (n+z)+1=n+(z+1)}\,(cut)}{\vdash n+(z+1)=(n+z)+1}\,\mathcal{E}
$$

and ψ_2

$$
\dfrac{\dfrac{\dfrac{\dfrac{\gamma_1}{\vdash z+(1+n)=(z+1)+n} \qquad \dfrac{\gamma_2}{\vdash 1+n=n+1}}{\vdash z+(n+1)=(z+1)+n}\,\mathcal{E}}{\vdash z+s(n)=(z+1)+n}\,A7}{\vdash s(z+n)=(z+1)+n}\,A4
$$

with γ_1

$$
\dfrac{\dfrac{\dfrac{\dfrac{\vdash z+1=z+1}{\vdash z+1=z+(1+0)}\,A3}{\vdash (z+1)+0=z+(1+0)}\,A3 \qquad \dfrac{\dfrac{\dfrac{\dfrac{(z+1)+u=z+(1+u)\vdash (z+1)+u=z+(1+u)}{(z+1)+u=z+(1+u)\vdash s((z+1)+u)=s(z+(1+u))}}{(z+1)+u=z+(1+u)\vdash s((z+1)+u)=z+s((1+u))}\,A4}{(z+1)+u=z+(1+u)\vdash s((z+1)+u)=z+(1+s(u))}\,A4}{(z+1)+u=z+(1+u)\vdash (z+1)+s(u)=z+(1+s(u))}\,A4}{\vdash (z+1)+n=z+(1+n)}\,(IND)}{\vdash z+(1+n)=(z+1)+n}\,\mathcal{E}
$$

and γ_2

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash 0 = 0}{\vdash 0 + 0 = 0}\,A3 \qquad \dfrac{\dfrac{\dfrac{\dfrac{0 + 0 = 0 \vdash 0 + 0 = 0}{0 + 0 = 0 \vdash s(0 + 0) = s(0)}}{0 + 0 = 0 \vdash 0 + s(0) = s(0)}\,A4}{0 + 0 = 0 \vdash 0 + 1 = 1}\,def}{\vdash 0 + 1 = 1}\,(cut)}{\vdash 0 + 1 = 1 + 0}\,A3 \qquad \dfrac{\tau}{w + 1 = 1 + w \vdash s(w) + 1 = 1 + s(w)}}{\dfrac{\vdash n + 1 = 1 + n}{\vdash 1 + n = n + 1}\,\mathcal{E}}\,(IND)$$

with τ

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{w + 1 = 1 + w \vdash w + 1 = 1 + w}{w + 1 = 1 + w \vdash s(w + 1) = s(1 + w)} \qquad \dfrac{\vdash 1 + s(w) = 1 + s(w)}{\vdash s(1 + w) = 1 + s(w)}\,A4}{w + 1 = 1 + w \vdash s(w + 1) = s(0) + s(w)}\,\mathcal{E}, def}{w + 1 = 1 + w \vdash s((w + 1) + 0) = s(0) + s(w)}\,A3}{w + 1 = 1 + w \vdash s(s(w) + 0) = s(0) + s(w)}\,A7}{w + 1 = 1 + w \vdash s(w) + s(0) = s(0) + s(w)}\,A4}{w + 1 = 1 + w \vdash s(w) + 1 = 1 + s(w)}\,def$$

Lemma 2. *Let π be a* ***PRA*** *proof of S. Then there exists a proof schema with end-sequent S.*

Proof. Let π contain α induction inferences

$$\dfrac{\Gamma_\beta \vdash \Delta_\beta, F_\beta(0) \qquad \Gamma_\beta, F_\beta(y) \vdash \Delta_\beta, F_\beta(y + 1)}{\Gamma_\beta \vdash \Delta_\beta, F_\beta(n_\beta)}\,(IND)$$

where $a \leq \beta \leq \alpha$. W.l.o.g. assume that if $\gamma < \beta$ then the induction inference with conclusion $\Gamma_\beta \vdash \Delta_\beta, F_\beta(n_\beta)$ is above the induction inference with conclusion $\Gamma_\gamma \vdash \Delta_\gamma, F_\gamma(n_\gamma)$. We define $\vec{n} = \{n_i \mid i \in \{1...\alpha\}\} \cup V(\pi)$ as the set of all induction variables, where n_i denotes the induction variable of the i-th induction inference, together with the set of free variables and constants V in π. Let T be the transformation taking an **PRA** proof to a proof schema by replacing the induction inferences with conclusion $\Gamma_\gamma \vdash \Delta_\gamma, F_\gamma(n_\gamma)$ by a proof call $(\delta_\gamma, \{m \leftarrow n_\gamma\}) : S(\delta_\gamma)\{m \leftarrow n_\gamma\}$ with $S(\delta_\gamma) = \Gamma_\gamma \vdash \Delta_\gamma, F_\gamma(m)$.

We will inductively construct a proof schema $\mathcal{D} = \{(\delta_i, \rho(\delta_i, \vec{n}, 0), \rho(\delta_i, \vec{n}, m + 1)) \mid i \in \{1...\alpha\}\}$ with end-sequent $S(\delta_i) = \Gamma_i \vdash \Delta_i, F_i(m + 1)$ for each tuple and $\delta_i > \delta_{i+1}$. Assume we already constructed proof schema $\mathcal{D}_{\beta+1} = \{(\delta_i, \rho(\delta_i, \vec{n}, 0), \rho(\delta_i, \vec{n}, m + 1)) \mid i \in \{(\beta + 1)...\alpha\}\}$.

Consider the induction inference with conclusion $\Gamma_\beta \vdash \Delta_\beta, F_\beta(n_\beta)$. Let φ_1 be the derivation above the left premise and φ_2 be the derivation above the right premise. We construct a proof schema $\mathcal{D}_\beta = \{(\delta_\beta, \rho(\delta_\beta, \vec{n}, 0), \rho(\delta_\beta, \vec{n}, m + 1))\} \cup \mathcal{D}_{\beta+1}$ with $\rho(\delta_\beta, \vec{n}, 0) = T(\varphi_1)$ and $\rho(\delta_\beta, \vec{n}, m + 1) =$

$$\dfrac{\dfrac{(\delta_\beta, \emptyset) : S(\delta_\beta)}{\Gamma_\beta \vdash \Delta_\beta, F_\beta(m)} \qquad \dfrac{T(\varphi_2)}{\Gamma_\beta, F_\beta(m) \vdash \Delta_\beta, F_\beta(m + 1)}}{\Gamma_\beta \vdash \Delta_\beta, F_\beta(m + 1)}\,(cut), (c : l)^*, (c : r)^*$$

Summarising, $(\delta_\beta, \rho(\delta_\beta, \vec{n}, 0), \rho(\delta_\beta, \vec{n}, m+1))$ is a proof schema tuple with end-sequent $\Gamma_\beta \vdash \Delta_\beta, F_\beta(n)$, as desired.

Finally, the part of π located beneath the last induction inference is translated into proof schema $\mathcal{D}' = \{(\delta', \rho(\delta', \vec{n}, m), \rho(\delta', \vec{n}, m+1))\} \cup \mathcal{D}$ with $S(\delta') = S$ and $\delta' > \delta_i$ for $i \in \{1...\alpha\}$. Let φ be the derivation above S. As there is no internal recursion in δ' needed, we only define $\rho(\delta', \vec{n}, m) = \dfrac{T(\varphi)}{S}$.

□

5 Conclusion

An interesting research question that has not been solved so far is to relate proof schemata to systems of arithmetic. In particular it was not known whether the classes of proofs specifiable in primitive recursive arithmetic and via proof schemata coincide. In this paper we have shown that proof schemata simulate primitive recursive arithmetic; we conjecture that a transformation of proofs in the other direction can be provided as well. Together with with a completeness result for the method CERES (a result not obtained so far) , the result in this paper yields a realization of Herbrand's theorem for an expressive fragment of formal number theory. While such a result would be far-fetched for general formula schemata, one has to take into account that formulas or clause sets derived from the cut structure of formal proofs are a significantly restricted subset of the set of all formulas or clause sets. Even if a completeness result for CERES on schemata originating from primitive recursive arithmetic were not possible using our current approach, we expect that a completeness result for an expressive fragment thereof is within reach. Such a result would be of major importance, as none of the previous schematic CERES methods is shown proof analytically complete for a fragment of primitive recursive arithmetic.

References

[1] Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter, and Hendrik Spohr. Ceres: An analysis of Fürstenberg's proof of the infinity of primes. *Theoretical Computer Science*, 403(2-3):160–175, 2008.

[2] Matthias Baaz and Alexander Leitsch. Cut-elimination and redundancy-elimination by resolution. *Journal of Symbolic Computation*, 29(2):149–177, 2000.

[3] James Brotherston. Cyclic proofs for first-order logic with inductive definitions. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 78–92. Springer, 2005.

[4] James Brotherston and Alex Simpson. Sequent calculi for induction and infinite descent. *Journal of Logic and Computation*, 21(6):1177–1216, 2011.

[5] David M. Cerna, Alexander Leitsch, and Anela Lolic. Schematic refutations of formula schemata. *J. Autom. Reason.*, 65(5):599–645, 2021.

[6] Cvetan Dunchev, Alexander Leitsch, Mikheil Rukhaia, and Daniel Weller. Cut-elimination and proof schemata. In *International Tbilisi Symposium on Logic, Language, and Computation*, pages 117–136. Springer, 2013.

[7] Jean Yves Girard. *Proof Theory and Logical Complexity*, volume 1. Biblopolis, Napoli,1987.

[8] Alexander Leitsch and Anela Lolic. Herbrand's theorem in refutation schemata. *arXiv preprint arXiv:2402.13905*, 2024.

[9] Alexander Leitsch, Nicolas Peltier, and Daniel Weller. CERES for first-order schemata. *J. Log. Comput.*, 27(7):1897–1954, 2017.

[10] Anela Lolic. *Automated Proof Analysis by CERES*. PhD thesis, Technical University of Vienna, 2020.

[11] Raymond McDowell and Dale Miller. Cut-elimination for a logic with definitions and induction. *Theoretical Computer Science*, 232(1-2):91–119, 2000.

[12] Gaisi Takeuti. *Proof Theory*. North Holland, second edition, 1987.