Proof Schemata for Peano Arithmetic

David M. Cerna[1,2]

[1] Research Institute for Symbolic Computation, Johannes Kepler University Linz, Austria
[2] Institute for Formal Models and Verification, Johannes Kepler University Linz, Austria
david.cerna@risc.jku.at

Abstract. Peano arithmetic (PA), as formalized by Gentzen, can be presented as an axiom extension of the **LK**-calculus with equality and an inference rule formalizing induction. While this formalism allowed for Gentzen's consistency argument, alternative formulations of induction, i.e. ω-arithmetic and cyclic reasoning, provide insight into the structure of arithmetic proofs obfuscated by Gentzen's inference formulation of induction. However, these variations require adding extra meta-level constructions which inhibit one's ability to perform certain computational proof transformations such as cut structure extraction and analysis. In this work we present a schematic sequent calculus for PA, which can be syntactically translated into the **LK**-calculus for PA while allowing cut structure extraction and analysis. One can think of this formalism as an explicit construction of ω-arithmetic restricted to primitive recursive (PR) sequences, i.e. the PR definitions are an object level constructions.

1 Introduction

Proof schemata serve as an alternative formulation of induction based on primitive recursive proof specification. Essentially, they are a collection of indexed derivations, so-called *proof components*, which can be linked together in a globally sound way resulting in a finitely describable infinite sequence of proofs. This construction is discussed in [8] where a calculus guaranteeing soundness is introduced for a fragment of arithmetic.

The seminal work concerning "proof as schema" was the analysis of Fürstenberg's proof of the infinitude of primes by Baaz et al. [2] using a rudimentary schematic formalism and **CERES** [3]. The schematic representation of induction aided the proof analysis by allowing one to consider a recursive representation of the cut structure as a set of clauses. From refutations of this recursive clause set and *proof projections* extracted of the pre-transformation proof one can extend Herbrand's theorem to a fragment of arithmetic [15], the same fragment discussed in [8].

This fragment restricts the introduction of fresh eigenvariables through induction. The corresponding restriction concerning proof schemata is the number of indices used is at most one. For example in [2] the index would be the number of primes assumed to exists. The $\mathcal{S}i$**LK**-calculus [8], internalizes this eigenvariable restrictions. Interestingly, the semantic interpretation of $\mathcal{S}i$**LK**-calculus sequents may be derived from Gentzen's transformation for fusing multiple inductions [11]. Removing this restriction on eigenvariable introduction while retaining the primitive recursive structure of the $\mathcal{S}i$**LK**-calculus is one of the novelties of this work (though we do not provide an inference system for the introduced formalism). In particular, $\mathcal{S}i$**LK**-calculus forms an important restriction of our formalism referred to as *computational proof schemata*.

There are several key distinctions between our new formalism and those found in literature. For example, the calculus introduced in [16] is closely related to the sequent calculus of Peano arithmetic, and was developed as a way to eliminate cut in the presence of induction without using the meta-theoretic method of Gentzen, however at the cost of analyticity of the resulting proof. Proof systems based on infinite descent [5, 6], otherwise known as cyclic proofs, maintain analyticity during the process of cut-elimination, though the resulting proof is no longer cyclic, nor regular, and thus, cannot be finitely described [21]. Similar results may be found concerning infinitely branching proof systems [7], even when considering restrictions to the ω-rule [19].

However, when the given cyclic proof is cut-free the finite traces of its regular infinite tree representation can be used to highlight a semantic relationships between two inductive definitions [17]. This is closely related to the resulting *Herbrand systems* of [15] which are extracted from a proof with cut by computational proof transformation techniques such as **CERES**, but only for a fragment of arithmetic.

Given the relationship between our formalism and proof transformation in the presence of induction, we make note of several proof transformation methods based on System T and the resulting functional interpretations of arithmetic [12], and the Friedman-A-translation [10]. While there are many works concerning such proof transformation techniques [1, 4, 18], either the techniques used are more analytic in nature, i.e. requiring mathematical ingenuity, or the techniques are computational but may only provide certain computational content, i.e. Π_2^0 statements. The system herein introduced was developed for **CERES** based computational proof transformation and thus allows for extraction of Π_1^0 computational content.

while one can imagine developing a **CERES** based transformation method for existing formalisms it has proven to be a non-trivial task. The same holds true for the translation of proofs from existing formalism into **CERES** friendly ones. Part of the difficultly with the development of such a formalism is that the standard formulation of primitive recursion is not sufficient for representation of proof schemata, it is too restrictive to allow the application of existing proof transformation methods. We use an alternative construction of primitive recursive functions based on the separation of variables into *active* and *dormant* [20]. This allows a more flexible proof construction and a syntactic translation to and from the Gentzen formulation of arithmetic. A corollary of this transformation is that our formalism is provability-wise as expressive as **PA**.

Our proof schemata are defined over an expressive well-founded ordering and allow multiple *free parameters* (eigenvariables introduced by inductive inferences) thus removing restrictions of earlier work [15]. Our Multi-parameter schemata allow quantifier introduction over numeric terms without the loss of the recursive structure and thus can formalize strong totality statements[3]. We leave for future work the extension of existing **CERES** methods to this formalism as well as using the formalism for proof transformation and computational content extraction.

2 The Schematic Language and P-schema

We generalize previous work concerning *proof schemata* [8, 9, 15]. Our formalism uses a multi-sorted first order language consisting of the ω sort, referred to as the *indexing sort*, and the ι sort, referred to as the *individual sort*. The ι sort represents the standard first-order term language together with a countable set of variable symbols. Given that this paper deals primarily with arithmetic, the indexing sort ω will be assumed to be the sort of numerals unless otherwise stated (we will refer to terms of this sort as numeric terms), that is terms of ω are constructed from the signature $\{0, s(\cdot)\}$ together with a countable set of *parameter symbols*. We assume no other function inhabit this sort unless otherwise stated. Parameters are special constants which can be thought of as *eigenvariables* for recursive proof definitions. We will discuss this in more detail when defining **P**-schema.

In addition to the above term constructions one can also define so-called *defined function symbols*, which can either be functions assigned a type constructed using the following grammar, $\tau := \omega|\iota|\tau \rightarrow \tau$, or primitive recursive definitions. Formula schemata are built from a countable set of predicate symbols $\mathcal{P}$, a countable set of defined predicate symbols $\hat{\mathcal{P}}$ and the logical connectives $\neg, \wedge, \vee, \rightarrow, \forall, \exists$. First-order formula which do not contain defined predicate symbols will be denoted by o.

Defined predicate symbols are typed predicate symbols. Their types are constructed using the grammar $\mu := \tau \rightarrow o$ where τ is the non-terminal introduced for typing defined function symbols. In general, defined symbols will be denoted by $\hat{\cdot}$, i.e. $\hat{P}$.

We assume a set of convergent rewrite rules $\mathcal{E}$ (equational theory) defining the syntactic behavior of defined symbols. The rules of $\mathcal{E}$ are of the form $\widehat{f}(\bar{t}) = E$, where $\bar{t}$ contains no defined symbols, and either $\widehat{f}$ is a defined function symbol and E is a term or $\widehat{f}$ is a defined predicate symbol and E is a formula schema. Rewrite rules of $\mathcal{E}$ may be applied to a sequent using the $\mathcal{E}$ inference rule. Not that this rule is reversible.

We generalize the notion of sequent to so-called *schematic sequents* which are a pair of multi-sets of formula schemata Δ, Π denoted by $\Delta \vdash \Pi$. We will

[3] As noted in [13], a single inductively introduced quantifier suffices for formalizing **PA**.

denote multi-sets of formula schemata by upper-case Greek letters. Let $S(\overline{x})$ be a schematic sequent and $\overline{x}$ a vector of free variables, then $S(\overline{t})$ denotes $S(\overline{x})$ where $\overline{x}$ is replaced by $\overline{t}$ and $\overline{t}$ is a vector of terms of appropriate type. We use the standard inference rules of the first-order **LK**-calculus (may be found in [22] or other standard proof theory literature) and extend them to handle the defined constructions of schematic sequents. We refer to this extended calculus as the **LKE**-calculus which is the **LK**-calculus with the addition of the following rule:

Definition 1 (LKE). *Let $\mathcal{E}$ be an equational theory. **LKE** is an extension of **LK** by the $\mathcal{E}$ inference rule* $\dfrac{S(t)}{S(t')}\,\mathcal{E}$ *where the term or formula schema t in the sequent S is replaced by a term or formula schema t' for $\mathcal{E} \models t = t'$.*

An **LKE**-derivation is a rooted tree s.t. every node is decorated by a sequent and an edge exists being two nodes of the tree iff there is a sound inference rule application with the sequent closer to the root playing the role of the main sequent and the other sequents playing the role of auxiliary sequents. If every branch of the tree ends at an initial sequent then we refer to the **LKE**-derivation as an **LKE**-proof. **P**-schema require an generalization of *initial sequents*, i.e. sequents of the form $A \vdash A$, which we cover in the following section.

2.1 The P-schema Formalism

To define **P**-schema we need to distinguish between three types of parameter symbols in the ω sort, namely, *active* parameters $\mathcal{N}_a$, *passive* parameters $\mathcal{N}_p$, and *internal* parameters $\mathcal{N}_i$[4]. These distinctions highlight three ways parameter symbols can be used within a proof schema:

- Allows proper construction of the recursive proof definition (active).
- Allows strong quantification, substitution, and proof normalization (passive).
- Allows value passing between different proofs within the **P**-schema (internal).

We will denote active parameters by lower-case Latin characters n, m, k, passive parameters by lower-case bold Greek characters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$, and internal parameters by lower-case bold Latin characters $\mathbf{n}, \mathbf{m}, \mathbf{k}$. Furthermore we distinguish between certain types of numeric terms by considering four independent sets of terms dependent on the type of parameter symbols they contain $\mathcal{A}_{\mathbb{N}}$ (active), $\mathcal{P}_{\mathbb{N}}$ (passive), $\mathcal{I}_{\mathbb{N}}$ (internal), and $\mathcal{G}_{\mathbb{N}}$ (parameter free). Together these sets make the set of *schematic terms* denoted by $\mathcal{S}_{\mathbb{N}} = \mathcal{A}_{\mathbb{N}} \cup \mathcal{P}_{\mathbb{N}} \cup \mathcal{G}_{\mathbb{N}} \cup \mathcal{I}_{\mathbb{N}}$. Given that we are concerned with terms built from successor and zero they contain at most one parameter symbol.

A particular type of schematic sequent is needed for the construction of **P**-schema which accounts for intended roles of the various types of parameters. We refer to a sequent S as $(n, \mathcal{I})$-sequent if the only active parameter occurring in S

[4] A related terminology can be found in [20] which discusses a construction similar to ours.

is n and all internal parameters occurring in S are members of the set $\mathcal{I} \subset \mathcal{I}_{\mathbb{N}}$. If a sequent S is active parameter free and all internal parameters occurring in S are members of the set $\mathcal{I} \subset \mathcal{I}_{\mathbb{N}}$ it will be referred to as an $\mathcal{I}$-sequent.

The presence or absence of an active parameter within a given sequent S denotes whether the sequent is intended to be used as part of the step case or base case of an inductive definition over $\mathcal{G}_{\mathbb{N}}$. If the sequent in question denotes the end sequent of an inductively defined **LK**-proof then the step case and base case must be syntactic variations of each other modulo the occurrences of the active parameter. We refer to a pair of sequents (S, S'), where S is an $(n, \mathcal{I})$-sequent and S' is an $\mathcal{I}$-sequent, as an *inductive pair* if there exists $\alpha \in \mathcal{G}_{\mathbb{N}}$ s.t. $S\{n \leftarrow \alpha\} = S'$.

Based on the parameter type distinction we can refine the concept of an **LKE**-derivation. We will refer to an **LKE**-derivation which only contains $(n, \mathcal{I})$-sequents as a *multivariate* **LKE**-derivation (*mv***LKE**-derivation). Furthermore an *mv***LKE**-derivation whose branches end at initial sequents and whose root sequent does not contain active and internal parameters is an *mv***LKE**-proof. The end-sequent of an *mv***LKE**-proof (*mv***LKE**-derivation) φ will be denoted by $es(\varphi)$ and the set $\mathcal{V}_x(S)$ for $x \in \{a, p, i\}$ will denote the active, passive and internal parameters occurring in the sequent S, respectively. Notice that the calculus introduced so far cannot construct *mv***LKE**-proofs unless the entire derivation is active and internal parameter free; **P**-schema describe how a set of derivations join to produce *mv***LKE**-proofs.

Non-axiomatic leaves of *mv***LKE**-derivations can be seen as *links* to a yet to be described proof. If one is given a set of proofs Φ containing a proof ψ which has a non-tautological leaf $S\sigma$, if there exists a $\chi \in \Phi$ s.t. $es(\chi) = S$ then we can *link* the proof χ to ψ to get a proof. The non-axiomatic leaves can be substitution invariants of the end sequent of χ. Note that σ is defined over the internal and active parameters of χ. If after a finite number of linking steps the result is a *mv***LKE**-derivation π, where each branch ends at an initial sequent, π can be easily transformed into an *mv***LKE**-proof by substitution.

In order to properly describe the above procedure we need to provide *mv***LKE**-derivations with names, therefore we assume a countably infinite set $\mathcal{B}$ of *proof symbols*, i.e. $\varphi, \psi, \varphi_i, \psi_j$.

Definition 2 (*mv*LKS). *The mv**LKS**-calculus is an extension of mv**LKE**, where links may appear at the leaves of a derivation.*

Given that *mv***LKS**-derivations contain links, an external object containing the derivations is necessary in order to guarantee a sound construction. This soundness condition is provided by the **P**-schema construction. To simplify some of the following definitions, we will refer to an *mv***LKS**-derivation as *inactive* if it does not contain an active parameter and $\{n\}$-active if it contains only the active parameter n.

Example 3. Consider the following $\mathcal{E}$ theory $\mathcal{E} = \{\widehat{a}(s(n), \beta) = s(\widehat{a}(n, \beta));\ \widehat{a}(0, \beta) = \beta\}$, where

$$\pi = \quad \dfrac{\dfrac{\vdash 0 = 0}{\vdash a(0,0) = 0}\,\mathcal{E}}{\vdash a(0,0) = a(0,0)}\,\mathcal{E} \qquad \nu = \quad \dfrac{\begin{array}{c}\chi(n)\\ \cdots\cdots\cdots\cdots\\ \vdash a(n,0) = a(0,n)\end{array} \qquad \begin{array}{c}\dfrac{S_1(\nu_1)}{\vdots}\end{array}}{\vdash a(n+1,0) = a(0,n+1)}\,cut$$

$S_1(\nu_1) \equiv a(n,0) = a(0,n) \vdash s(a(n,0)) = s(a(0,n))$, π is an mv**LKS**-proof and ν is an mv**LKS**-derivation. Also, ν contains a link to the proof symbol χ and is $\{n\}$-active. Note that the end-sequent of ν, $es(\nu)$ is $es(\chi(n))\{n \leftarrow n+1\}$. Moreover, $es(\nu)$ is an $(n,\emptyset)$-sequent, i.e. internal parameter free. Its inductive pair also contains the sequent $es(\nu)\{n \leftarrow 0\} = \vdash a(0,0) = a(0,0)$, the end-sequent of π (an $\emptyset$-sequent). The triple (χ, π_1, ν_1) is referred to as an $(n,\emptyset)$-component.

Definition 4 ($(n,\mathcal{I})$-component). *Let $\psi \in \mathcal{B}$, $n \in \mathcal{N}_a$ and $\mathcal{I} \subset \mathcal{N}_i$. An $(n,\mathcal{I})$-*component **C** *is a triple (ψ,π,ν) where π is an inactive mv**LKS**-derivation ending with $S\{n \leftarrow \alpha\}$ and ν is an $\{n\}$-active mv**LKS**-derivation ending in an $(n,\mathcal{I})$-sequent S whose inductive pair is $S\{n \leftarrow \alpha\}$. Given a component $\mathbf{C} = (\psi,\pi,\nu)$ we define $\mathbf{C}.1 = \psi$, $\mathbf{C}.2 = \pi$, and $\mathbf{C}.3 = \nu$. We refer to $es(\mathbf{C}) = S$ as the* end sequent *of the component* **C**.

When possible, we will refer to an $(n,\mathcal{I})$-component as a component. A schematic proof is defined over finitely many components, which can be linked together. So far we have not restricted linking, not all links are sound. Whenever a component **C** links to another component **D** the passive parameters occurring in $es(\mathbf{D})$ must occur in the sequent associated with the link in **C**, what we refer to as *association*. In order to define associations between schematic sequents we introduce *schematic substitutions*, a function $\sigma : \mathcal{A}_{\mathbb{N}} \cup \mathcal{I}_{\mathbb{N}} \rightarrow \mathcal{S}_{\mathbb{N}}$ which replaces all occurrences of a parameter $x \in \mathcal{A}_{\mathbb{N}} \cup \mathcal{I}_{\mathbb{N}}$ with a term $t \in \mathcal{S}_{\mathbb{N}}$.

Definition 5 (association). *Let $S(\bar{t})$ be an $(n,\mathcal{I})$-sequent and $S'(\bar{x})$ an $(m,\mathcal{I}')$-sequent where $\bar{t}$ is a sequence of terms from the ι sort and $\bar{x}$ is a sequence of free ι sort variables of the same length. We say $S(\bar{t})$* associates with *$S'(\bar{x})$ if there exists a schematic substitution σ s.t. $S(\bar{t})\sigma = S'(\bar{t})$.*

Association defines the relationship between schematic sequents with roughly the same structure, however, for **P**-schema we need a slightly stronger relation which we refer to as *Linkability.* Essentially, linkability tells us when the end sequent of a given component can be attached to leaves of another component, i.e. the two sequents associate and certain restrictions on the passive parameters hold.

Definition 6 (Linkability). *Two components* **C** *and* **D** *are said to be* $(\mathbf{C},\mathbf{D})$-linkable *if for each non-axiomatic leaf S in* **C** *of which $es(\mathbf{D})$ associates, $\mathcal{V}_p(es(\mathbf{D})) \subseteq \mathcal{V}_p(S)$. We say they are* strictly $(\mathbf{C},\mathbf{D})$-linkable *if it holds that $\mathcal{V}_p(es(\mathbf{D})) \subseteq \mathcal{V}_p(es(\mathbf{C}))$.*

Assuming we have a set of components Φ, linkability defines a partial order.

Definition 7 (Linkability ordering). *Let $\mathbf{C}_1$ and $\mathbf{C}_2$ be distinct components s.t. they are (strictly) $(\mathbf{C}_1,\mathbf{C}_2)$-linkable. Then we say that $\mathbf{C}_1 \prec \mathbf{C}_2$ ($\mathbf{C}_1 \prec_s \mathbf{C}_2$).*

Our restriction on the number of active parameters per sequent blocks mutual recursion and linkability relation can be used to define a well-ordering thus enforcing a primitive recursive construction. However, the linkability ordering is not well-founded, it is defined over the set of all components and allows the definition of mutually linkable components, i.e. $(\mathbf{C}_1, \mathbf{C}_2)$-linkable and $(\mathbf{C}_2, \mathbf{C}_1)$-linkable. Thus, we further restrict our usage of the ordering and only consider well founded sub-orderings of the linkability ordering when defining **P**-schemata.

Definition 8 (P-schema). *Let* $\mathbf{P} \subset \mathcal{N}_p$, $\mathbf{C}_1$ *an* $(n, \mathcal{I})$*-component and* $\mathbf{C}_2, \cdots, \mathbf{C}_\alpha$ *components s.t. for all* $1 \leq i \leq \alpha$, $\mathbf{C}_i.1$ *are distinct and* $\mathcal{V}_p(\mathbf{C}_i) \subseteq \mathbf{P}$. *We define* $\Psi = \langle \mathbf{C}_1, \cdots, \mathbf{C}_\alpha \rangle$ *as a* **P**-schema *(*strict **P**-schema*) over a well founded suborder* $\prec^* \subset \prec$ *(* $\prec^*_s \subset \prec_s$*) of* $\{\mathbf{C}_1, \ldots, \mathbf{C}_\alpha\}$ *with* $\mathbf{C}_1$ *as least element. We define* $|\Psi| = \alpha$, $\Psi.i = \mathbf{C}_i$ *for* $1 \leq i \leq \alpha$, *and* $es(\Psi) = es(\mathbf{C}_1)$.

Notice that the passive parameters are, in some sense, declared before the construction of the **P**-schema, i.e. **P** is a set of passive parameters. These symbols play a special role during the normalization procedure, i.e. when we unfold the primitive recursive definitions defining a **P**-schema and construct an **LKS**-Proof. One can think of passive parameters as constants acting as place holders for numerals, i.e. ground terms $\mathcal{G}_\mathbb{N}$. However, not all passive parameters in a **P**-schema need to occur in the end sequent of a given schema. Passive parameters can also play the role of numeric eigenvariables and be strongly quantified. This property enforces a special treatment of **P**-schema which are constructed from other **P**-schema.

Definition 9 (sub P-schema). *Let* $\Psi = \langle \mathbf{C}_1, \cdots, \mathbf{C}_\alpha \rangle$ *be a* **P***-schema and* $\Psi' = \left\langle \mathbf{C}'_1, \cdots, \mathbf{C}'_\beta \right\rangle$ *be a* **P***-schema s.t.* $\left\{ \mathbf{C}'_1, \cdots, \mathbf{C}'_\beta \right\} \subseteq \{\mathbf{C}_1, \cdots, \mathbf{C}_\alpha\}$. *We refer to* Ψ' *as a* sub **P**-schema *of* Ψ. *Furthermore, consider a* **P***-schema* $\Phi = \left\langle \mathbf{C}, \mathbf{C}'_1, \cdots, \mathbf{C}'_\beta \right\rangle$, *a component* $C' \in \Psi'$ *s.t. for all* $D \in \Psi'$, *if* $D \neq C'$, *then* $C' \not\prec D$ *and an occurrence of a link to* C' *in* Ψ *of the form* $\dfrac{C'.1(t, \cdots)}{es(C'.1)\sigma}$ *Where* σ *is a substitution with* $dom(\sigma) = \mathcal{V}_i(C') \cup \mathcal{V}_a(C')$ *s.t.* $\mathcal{V}_a(C')\sigma = \{t\}$. *We refer to a sub* **P***-schema as* ground *if* $\mathcal{V}_a(t) = \mathcal{V}_i(t) = \mathcal{V}_p(t) = \emptyset$, essential *if* $\mathcal{V}_a(t) \neq \emptyset$ *and* $\mathcal{V}_p(t) = \emptyset$, *or as* computational *if* $\mathcal{V}_a(t) = \emptyset$ *and* $\mathcal{V}_p(t) \neq \emptyset$, *but* $\mathcal{V}_p(t) \cap \mathcal{V}_p(es(\Psi)) = \emptyset$.

Example 10. We can formalize associativity of addition as an $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\}$-schema $\Phi = \langle (\varphi, \pi, \nu) \rangle$ over the following $\mathcal{E}$ theory $\mathcal{E} = \{\widehat{a}(s(n), \beta) = s(\widehat{a}(n, \beta));\ \widehat{a}(0, \beta) = \beta\}$ where

$$\pi = \cfrac{\cfrac{\vdash a(\mathbf{k}, \boldsymbol{\gamma}) = a(\mathbf{k}, \boldsymbol{\gamma})}{\vdash a(0, a(\mathbf{k}, \boldsymbol{\gamma})) = a(\mathbf{k}, \boldsymbol{\gamma})}\ \mathcal{E}}{\vdash a(0, a(\mathbf{k}, \boldsymbol{\gamma})) = a(a(0, \mathbf{k}), \boldsymbol{\gamma})}\ \mathcal{E} \qquad \nu = \cfrac{\begin{array}{c}\varphi(n, \mathbf{k}, \boldsymbol{\gamma}) \\ \vdots \\ \vdash a(n, a(\mathbf{k}, \boldsymbol{\gamma})) = a(a(n, \mathbf{k}), \boldsymbol{\gamma})\end{array} \quad \begin{array}{c}(\nu_1) \\ \\ S(\nu_1)\end{array}}{\vdash a(n', a(\mathbf{k}, \boldsymbol{\gamma})) = a(a(n', \mathbf{k}), \boldsymbol{\gamma})}\ cut$$

and $S(\nu_1) \equiv a(n, a(\mathbf{k}, \boldsymbol{\gamma})) = a(a(n, \mathbf{k}), \boldsymbol{\gamma}) \vdash a(n', a(\mathbf{k}, \boldsymbol{\gamma})) = a(a(n', \mathbf{k}), \boldsymbol{\gamma})$. Notice that ν is an *mv***LKS**-derivation not an *mv***LKS**-proof being that the end sequent of ν is $\{n\}$-active. We can extend Φ to $\Phi^* = \langle (\chi, \lambda, \mu), (\varphi, \pi, \nu) \rangle$ where

$$\nu_1 = \dfrac{\dfrac{\dfrac{a(n,a(\mathbf{k},\boldsymbol{\gamma})) = a(a(n,\mathbf{k}),\boldsymbol{\gamma}) \vdash s(a(n,a(\mathbf{k},\boldsymbol{\gamma}))) = s(a(a(n,\mathbf{k}),\boldsymbol{\gamma}))}{a(n,a(\mathbf{k},\boldsymbol{\gamma})) = a(a(n,\mathbf{k}),\boldsymbol{\gamma}) \vdash a(n',a(\mathbf{k},\boldsymbol{\gamma})) = s(a(a(n,\mathbf{k}),\boldsymbol{\gamma}))}\,\mathcal{E}}{a(n,a(\mathbf{k},\boldsymbol{\gamma})) = a(a(n,\mathbf{k}),\boldsymbol{\gamma}) \vdash a(n',a(\mathbf{k},\boldsymbol{\gamma})) = a(s(a(n,\mathbf{k})),\boldsymbol{\gamma})}\,\mathcal{E}}{a(n,a(\mathbf{k},\boldsymbol{\gamma})) = a(a(n,\mathbf{k}),\boldsymbol{\gamma}) \vdash a(n',a(\mathbf{k},\boldsymbol{\gamma})) = a(a(n',\mathbf{k}),\boldsymbol{\gamma})}\,\mathcal{E}$$

$$\lambda = \begin{array}{c} \varphi(0,\boldsymbol{\beta},\boldsymbol{\gamma}) \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \vdash a(0,a(\boldsymbol{\beta},\boldsymbol{\gamma})) = a(a(0,\boldsymbol{\beta}),\boldsymbol{\gamma}) \end{array}$$

$$\mu = \begin{array}{c} \varphi(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \vdash a(\boldsymbol{\alpha},a(\boldsymbol{\beta},\boldsymbol{\gamma})) = a(a(\boldsymbol{\alpha},\boldsymbol{\beta}),\boldsymbol{\gamma}) \end{array}$$

The schema Φ^* ends with an mv**LKS**-proof and represents a sequence of mv**LKS**-proofs.

Note that this formalization is a generalization of the formalization described in [15]. If we were to restrict ourselves to $(n,\mathcal{I})$-components and construct strict $\{\boldsymbol{\alpha}\}$-schema, the resulting formalism would be equivalent to proof schemata *á la* [15]. For example $\{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}\}$-schema Φ provided in Example 10 has an $\{n\}$-active end sequent with a free internal parameter, these are nothing more than the free parameter and a free variable of the ι sort as discussed in [15]. We can extend this example to a proof of commutativity which is beyond the expressive power of previous formalizations.

Example 11. We use the same $\mathcal{E}$ theory as presented in Example 10 and extend the $\{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}\}$-schema of Example 10 to the $\{\boldsymbol{\alpha},\boldsymbol{\beta}\}$-schema $\Phi' = \langle(\chi,\pi_1,\nu_1),(\psi,\pi_2,\nu_2),(\xi,\pi_3,\nu_3)(\varphi,\pi,\nu)\rangle$ using the following equational axioms:

$$E_1 \equiv a(\boldsymbol{\alpha},1)=a(1,\boldsymbol{\alpha}) \vdash a(a(\boldsymbol{\alpha},1),n)=a(a(1,\boldsymbol{\alpha}),n)$$
$$E_2 \equiv a(a(1,\boldsymbol{\alpha}),n)=a(n',\boldsymbol{\alpha}),a(a(\boldsymbol{\alpha},1),n)=a(a(1,\boldsymbol{\alpha}),n) \vdash a(a(\boldsymbol{\alpha},1),n)=a(n',\boldsymbol{\alpha})$$
$$E_3 \equiv a(\boldsymbol{\alpha},n)=a(n,\boldsymbol{\alpha}) \vdash s(a(\boldsymbol{\alpha},n))=s(a(n,\boldsymbol{\alpha}))$$
$$E_4 \equiv a(\boldsymbol{\alpha},a(1,n))=a(a(\boldsymbol{\alpha},1),n),a(a(\boldsymbol{\alpha},1),n)=a(n',\boldsymbol{\alpha}) \vdash a(\boldsymbol{\alpha},a(1,n))=a(n',\boldsymbol{\alpha})$$

π_1 and ν_1 are as in Example 3.

$$\pi_2 = \dfrac{\dfrac{\dfrac{\vdash s(0) = s(0)}{\vdash a(0,s(0)) = s(0)}}{\vdash a(0,s(0)) = s(a(0,0))}\,\mathcal{E}}{\vdash a(0,s(0)) = a(s(0),0)}\,\mathcal{E}$$

$$\nu_2 = \dfrac{\dfrac{\begin{array}{c}\psi(n)\\ \cdots\cdots\cdots \\ \vdash a(n,1) = a(1,n)\end{array} \quad S_1(\nu_2)}{\vdots}\,cut}{\vdash a(n',1) = a(1,n')}$$

where $S_1(\nu_2) \equiv a(n,1) = a(1,n) \vdash s(a(n,1)) = s(a(1,n))$, and

$$\pi_3 = \begin{array}{c} \chi(\boldsymbol{\alpha}) \\ \cdots\cdots\cdots \\ \vdash a(\boldsymbol{\alpha},0) = a(0,\boldsymbol{\alpha}) \end{array}$$

$$\nu_{3_1} = \begin{array}{c} \varphi(\boldsymbol{\alpha},1,n) \\ \cdots\cdots\cdots\cdots \\ \vdash a(\boldsymbol{\alpha},a(1,n)) = a(a(\boldsymbol{\alpha},1),n) \end{array}$$

$$\nu_3 = \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\xi(n,\boldsymbol{\alpha})\\ \cdots\cdots\cdots \\ \vdash a(\boldsymbol{\alpha},n) = a(n,\boldsymbol{\alpha})\end{array}}{\vdots} \quad \dfrac{\begin{array}{c}\psi(\boldsymbol{\alpha})\\ \cdots\cdots\cdots \\ \vdash a(\boldsymbol{\alpha},1) = a(1,\boldsymbol{\alpha})\end{array}}{\vdots}}{\vdots}\,cut \quad \begin{array}{c}(\nu_{3_1})\\ S(\nu_{3_1})\end{array}}{\vdash a(\boldsymbol{\alpha},a(1,n)) = a(n',\boldsymbol{\alpha})}\,cut}{\vdash a(\boldsymbol{\alpha},s(a(0,n))) = a(n',\boldsymbol{\alpha})}\,\mathcal{E}}{\vdash a(\boldsymbol{\alpha},n') = a(n',\boldsymbol{\alpha})}\,\mathcal{E}$$

where $S(\nu_{3_1}) \equiv \vdash a(\boldsymbol{\alpha},a(1,n)) = a(a(\boldsymbol{\alpha},1),n)$. Notice that ξ is the least element of the order $\prec$ and the following relations concerning $\prec$ are also defined: $\xi \prec \varphi$, $\xi \prec \psi$, $\psi \prec \chi$. Note Evaluation of Φ' into an **LKS**-proof is not yet possible given that $es(\Phi')$ contains an active parameter. We can perform a similar extension as before to construct a **P**-schema which can be evaluated. We can now quantify the passive parameters of the schema and derive the precise statement of commutativity as one would derive in **PA**.

Constructing an *mv***LKS**-proof from an *mv***LKS**-derivation is defined as follows:

Theorem 12. *Let $\Phi = \langle \mathbf{C}_1, \cdots, \mathbf{C}_\alpha \rangle$ be a* **P***-schema resulting in an mv**LKS**-derivation s.t.* $0 < |\mathcal{V}_i(es(\Phi))| + |\mathcal{V}_a(es(\Phi))|$ *and* $\mathcal{V}_p(es(\Phi))| + |\mathcal{V}_i(es(\Phi))| + |\mathcal{V}_a(es(\Phi))| \leq |P|$*. Then there exists* $\Phi' = \langle \mathbf{C}^*, \mathbf{C}_1, \cdots, \mathbf{C}_\alpha \rangle$ *s.t.* $0 = |\mathcal{V}_i(es(\Phi))| + |\mathcal{V}_a(es(\Phi))|$*.*

Proof. We add a new component to Φ resulting in the proof schema $\Phi' = \langle \mathbf{C}^*, \mathbf{C}_1, \cdots, \mathbf{C}_\alpha \rangle$ s.t. $\mathbf{C}_1 \prec \mathbf{C}^*$. Let $es(\Phi) = S(n, \boldsymbol{\alpha}_1, \cdots \boldsymbol{\alpha}_k, \mathbf{i}_1, \cdots, \mathbf{i}_m)$, $\mathbf{C}^* = (\chi, \lambda, \mu)$ and $\mathbf{C}_1 = (\varphi, \lambda', \mu')$. Furthermore, let $S = \{\gamma, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_m\} \subset \mathbf{P}$ s.t. $\{\boldsymbol{\alpha}_1, \cdots \boldsymbol{\alpha}_k\} \cup S \equiv \emptyset$. Then we define λ and μ as follows

$$\lambda = \frac{\varphi(0, \boldsymbol{\alpha}_1, \cdots \boldsymbol{\alpha}_k, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_m)}{S(0, \boldsymbol{\alpha}_1, \cdots \boldsymbol{\alpha}_k, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_m)} \qquad \mu = \frac{\varphi(\gamma, \boldsymbol{\alpha}_1, \cdots \boldsymbol{\alpha}_k, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_m)}{S(\gamma, \boldsymbol{\alpha}_1, \cdots \boldsymbol{\alpha}_k, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_m)}$$

□

We will refer to such a **P**-schema as the *completion* of Φ, or a *complete* **P**-schema. In the next section we show that evaluation and soundness [15] can be extended to our formalism.

2.2 Evaluating P-Schemata

We now extend the soundness result of proof schema [15] and define evaluation of **P**-schema.

Definition 13 (Evaluation of P-schemata). *Let $\Phi = \langle \mathbf{C}_1, \cdots, \mathbf{C}_\alpha \rangle$ be a complete strict* **P***-schema s.t. $C_i = (\psi_i, \pi_i, \nu_i)$. We define the rewrite rules for links as $\hat{\psi}_i(0, \mathcal{I}) \rightarrow \pi_i$ and $\hat{\psi}_i(s(n), \mathcal{I}) \rightarrow \nu_i$ where $\mathbf{C}_i$ is assumed to contain the active parameter n. The rewrite system for links is the union of these rules. Moreover, for a substitution $\sigma : \mathcal{N}_p \rightarrow \mathcal{G}_{\mathbb{N}}$ with domain P, we define $\Phi\sigma = \hat{\psi}_1\sigma$ as the normal form of Φ under these rewrite rules and the equational theory $\mathcal{E}$.*

Lemma 14. *Let Φ be a complete strict* **P***-schema and $\sigma : \mathcal{N}_p \rightarrow \mathcal{G}_{\mathbb{N}}$ with domain P. The rewrite system for the links of Φ is strongly normalizing and confluent, s.t. $\Phi\sigma$ is an **LK**-proof.*

Proof. By the restriction on occurrences of links, a proof schema can be seen as a set of primitive recursive definitions, and the rewrite rules for links are the standard rules for these definitions. It is well-known that such rewrite systems are strongly normalizing, see [14]. Finally, by the restriction to complete strict **P**-schema links will not occur in the normal form and $\Phi\sigma$ is an *mv***LKE**-proof. Furthermore, since all $\mathcal{E}$-inferences in this proof are trivial and there are no parameters, we may consider it as an **LK**-proof.

□

The conditions we place on **P**-schema are sufficient for the development of a strongly normalizing and confluent rewrite system, but are not necessary. As has been pointed out we can relax these conditions by enforcing the components to obey a well-founded order. Relaxing our conditions in this way would of course yield a more flexible formalism allowing the occurrence of arbitrary active parameters within a given schematic sequent, but at the same time the complexity of the formalism would have to increase. What is interesting about this work is that none of this complexity is necessary in order to achieve provability equivalence to expressive arithmetic theories.

Though there is still some room for us to relax the constraints. In particular we can allow proof schema to have computational sub-**P**-schema. starting from the sub-**P**-schema of a **P**-schema Φ which do not have any computational sub-**P**-schema we can apply Lemma 14 and construct an **LK**-proof. By structural induction on the **P**-schema construction we can consider the next largest computational sub-**P**-schema and glue the unfolded **P**-schema together and construct a larger proof. Though, we are most interested in complete strict **P**-schema and their relation to primitive recursive arithmetic. By theory extension, we mean any link to a sub **P**-schema can be treated as an axiom because it can be unrolled into an **LK**-proof and is therefore soundly constructed. Before discussing the soundness of **P**-schemata with respect to **LK**-proofs we need to introduce an additional concept concerning computational sub-**P**-schema.

Definition 15. *Let Ψ be a* **P***-schema. We refer to a* **P***-schema χ as a* direct *sub-***P***-schema of Ψ if there does not exists a sub-***P***-schema ξ other than Ψ itself s.t. χ is also a sub-***P***-schema of ξ and $\chi \not\equiv \Psi$.*

Definition 16. *Let Ψ be a* **P***-schema. We will refer to Ψ as 0-computational if Ψ is a complete* **P***-schema and does not contain any computational sub* **P***-schema. We refer to Ψ as k-computational if there exists a direct sub-***P***-schema of Ψ which is $(k-1)$-computational.*

Theorem 17 (Soundness of P-schemata). *Let Ψ be a* **P***-schema and let $\sigma : \mathcal{N}_p \rightarrow \mathcal{G}_{\mathbb{N}}$ be a substitution whose domain is* **P***. Then $\Psi\sigma$ is an* ***LK****-proof over a theory T of $es(\Psi)\sigma$, where T is the set of end-sequents of the direct computational sub* **P***-schema Ψ.*

Proof. Let us assume that Ψ is a 0-computational **P**-schema implying that Ψ does not contain any computational sub-**P**-schema and that $es(\Psi)$ does not contain internal, nor active parameters. Thus by Lemma 14, $es(\Psi)\sigma$ can be transformed into an **LK**-proof.

For the induction hypothesis, we assume that for any k-computational **P**-schema Ψ, $\Psi\sigma$ can be transformed into a **LK**-proof over a theory T consisting of the end sequents of the direct sub **P**-schema of Ψ. We show that for a $(k+1)$-computational **P**-schema Ψ', $\Psi'\sigma'$ can be transformed into a **LK**-proof over a theory T' consisting of the end sequents of the direct sub **P**-schema of Ψ'. Let S be the set of all direct sub-**P**-schema of Ψ'.

Each $\chi \in S$ is an m-computational **P**-schema for $m \leq k$ and thus, we know that χ can be transformed into a complete **P**-schema and there exists a substitution σ_χ s.t. $\chi\sigma_\chi$ is an **LK**-proof over some theory T consisting of the end sequents of the direct sub-**P**-schema of χ. Let the end sequent of $\chi\sigma_\chi$ by $\Delta \vdash \Pi$. We can easily extend $\chi\sigma_\chi$ to a proof of $S_\chi = \vdash \bigwedge_{\phi \in \Delta} \phi \rightarrow \bigvee_{\phi \in \Pi} \phi$. Finally, this implies that we can construct a proof $\vdash \bigwedge_{\chi \in S'} S_\chi$. Thus, the conjunction of the end sequents of each sub-**P**-schema in $\chi \in S'$ is a sound construction and an **LK**-proof.

We can consider the Ψ^* which is the **P**-schema derived from Ψ by removing all components associated with sub-**P**-schema of S'. Occurrences of the end sequents of sub-**P**-schema of S' within Ψ^* can be treated as theory and thus $\Psi^*\sigma'$ is an **LK**-proof modulo the theory consisting of the end sequents of the computational direct sub **P**-schema of Ψ'. □

3 Local Induction and *mv*LKE

In [15], it was shown that proof schemata are equivalent to a particular fragment of arithmetic, i.e. the so-called *k-simple induction*, which limits the introduction of fresh eigenvariables by induction. The induction rule on the left-hand side:

$$\frac{F(k), \Gamma \vdash \Delta, F(s(k))}{F(0), \Gamma \vdash \Delta, F(t)}\ \mathbf{IND} \qquad\qquad \frac{F(n), \Gamma \vdash \Delta, F(n+1)}{F(0)\sigma, \Gamma\sigma \vdash \Delta\sigma, F(t)\sigma}\ mv\mathbf{IND}$$

where t is a term of the numeric sort s.t. t either contains k (k is a free parameter in the sense of [15]) or is ground. Adding the above rule to **LKE** resulted in the **LKIE**-calculus. We will refer to this calculus as the *simple* **LKIE**-calculus. The *mv***LKE**-calculus and **P**-schema are related to a much more expressive induction rule. Essentially any term t can replace the active parameter n of the auxiliary sequent (including a term containing n) and the internal parameters can be instantiated with arbitrary terms. The instantiations must obey the restriction of at most one active parameter per schematic sequent. We refer to the calculus with the right-hand side induction rule above as the *mv***LKIE**-calculus, where σ is a substitution whose domain is equivalent to the internal parameters occurring in the auxiliary sequent of the *mv***IND** induction inference. Note that internal parameters behave like free variables and thus can be instantiated by any term. In some sense they are internal with respect to a given component and thus, as we shall see later, are superfluous. We leave them as part of the formalism given that it is sensible for recursive definition to pass information. We do not restrict the terms σ can map internal parameters to other than the mapping must obey our restriction on active parameters, that is, only one active parameter per schematic sequent.

We consider an *mv***LKIE**-derivation ψ as an *mv***LKIE**-proof if the end-sequent of ψ does not contain internal or active parameters. We will first consider strict *mv***LKIE**-proofs which, like strict links, require preservation of the passive parameters, i.e. all passive parameters used in the proof must show up in the end sequent of the proof.

Example 18. Here we present a strict mv**LKIE**-proof of the $\{\boldsymbol{\alpha},\boldsymbol{\beta}\}$-schema of Example 11. Notice how the links are replaced by the induction rules in a similar fashion to the k-induction conversion introduced in [15]. We will further discuss this conversion with respect to **P**-schema later in this paper.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \dfrac{\begin{array}{c}a(n,1)=a(1,n)\vdash\\ a(n',1)=a(1,n')\end{array}}{\begin{array}{c}a(0,1)=a(1,0)\vdash\\ a(\boldsymbol{\alpha},1)=a(1,\boldsymbol{\alpha})\end{array}}\,mv\mathbf{IND}\end{array}\qquad \begin{array}{c}\dfrac{\begin{array}{c}\vdots\\ a(n,a(\mathbf{m},\mathbf{k}))=a(a(n,\mathbf{m}),\mathbf{k})\vdash\\ a(n',a(\mathbf{m},\mathbf{k}))=a(a(n',\mathbf{m}),\mathbf{k})\end{array}}{\begin{array}{c}a(0,a(1,n))=a(a(0,1),n)\vdash\\ a(\boldsymbol{\alpha},a(1,n))=a(a(\boldsymbol{\alpha},1),n)\end{array}}\,mv\mathbf{IND}\\ \vdots\\ \vdash a(\boldsymbol{\alpha},a(1,n))=a(a(\boldsymbol{\alpha},1),n)\end{array}}{\begin{array}{c}a(\boldsymbol{\alpha},n)=a(n,\boldsymbol{\alpha})\vdash\\ a(\boldsymbol{\alpha},a(1,n))=a(n',\boldsymbol{\alpha})\end{array}}\,cut}{\begin{array}{c}a(\boldsymbol{\alpha},n)=a(n,\boldsymbol{\alpha})\vdash\\ a(\boldsymbol{\alpha},s(a(0,n)))=a(n',\boldsymbol{\alpha})\end{array}}\,\mathcal{E}}{\begin{array}{c}a(\boldsymbol{\alpha},n)=a(n,\boldsymbol{\alpha})\vdash\\ a(\boldsymbol{\alpha},n')=a(n',\boldsymbol{\alpha})\end{array}}\,\mathcal{E}}{\begin{array}{c}a(\boldsymbol{\alpha},0)=a(0,\boldsymbol{\alpha})\vdash\\ a(\boldsymbol{\alpha},\boldsymbol{\beta})=a(\boldsymbol{\beta},\boldsymbol{\alpha})\end{array}}\,mv\mathbf{IND}
$$

Syntactically transforming **LKIE**-proofs into proof schemata, as done in [15], does not directly address the k-simplicity of the induction inferences. Rather, the transformation required a structural similarity between the usage of the indexing sort of the proof schema and the induction inferences in the **LKIE**-proof. The k-simple induction inference provides a perfect correspondence with proof schema as defined in [15]. Thus, if we are to follow a similar argumentation, we need to show that the mv**IND** induction inference captures **P**-schema. This argument is easier to make when we restrict ourselves to strict proofs, i.e. transforming strict mv**LKIE**-proofs into complete strict **P**-schema. However, using a similar argumentation as we used in Theorem 17, one can generalize this translation to a method for translating mv**LKIE**-proofs into complete **P**-schema.

Lemma 19. *For strict (when $\mathcal{V}_p(es(\Psi)))\equiv P$)* **P***-schema Ψ there exists a strict mv**LKIE**-derivation of $es(\Psi)$.*

Proof. Let us consider a strict $\{\boldsymbol{\alpha}\}$-schema Ψ. As we discussed earlier such **P**-schema are equivalent to the proof schemata defined in [15]. We repeat the argument provided in the earlier work translating strict $\{\boldsymbol{\alpha}\}$-schema into strict mv**LKIE**-derivation of $es(\Psi)$.

Let us consider all $C\in\Psi$ s.t. for any $D\in\Psi$ if $C\neq D$, then $C\not\prec D$. Furthermore, let us consider the sub-$\{\boldsymbol{\alpha}\}$-schema of Ψ, Φ. We show how one can construct a mv**LKIE**-derivation for such a component C. Being that C can only link to itself and we are restricted to a single passive parameter, it is only necessary to consider proof links in $C.3$. Now we produce a proof ν from $C.3$ by replacing all links of the form

$$
\begin{array}{c} C.1(n,t_1,\cdots,t_n) \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ es(C.1)(n,t_1,\cdots,t_n) \end{array}
$$

by the schematic sequent $\mathcal{F}(es(C.1))\sigma\vdash\mathcal{F}(es(C.1))\sigma$ where σ is a substitution implementing the substitution of the link, i.e. $\sigma:\mathcal{V}_i(es(C.1))\to\mathcal{S}_{\mathbb{N}}$. There may be more then one occurrence of a link to $C.1$ within $C.3$ and thus multiple substitutions may occur. After applying this transformation the end sequent of ν becomes $\mathcal{F}(es(C.1)) \vdash \mathcal{F}(es(C.1))\,\{n \leftarrow n+1\}$. The resulting end sequent can be used as an auxiliary sequent for an mv**IND** inference. So far we only considered self referential links, thus we do not need to consider substitutions for the internal parameters, thus the mv**IND** inference behaves like a **IND** inference. In the step case there will be a slight complication arising when a link occurs pointing to another proof symbol. In such cases internal parameters can be instantiated as part of the mv**IND** inferences. Thus, when using this proof as a sub-proof of a larger proof one must use the appropriate substitution. To complete the construction we apply a cut inference to the main sequent of the mv**IND** inference and the end sequent of $C.2$, i.e. the basecase.

For the step case let $C \in \Psi$ s.t. for all $D_1, \cdots, D_m \in \Psi$ of which $D_m \prec \cdots \prec D_1 \prec C$ it must be the case that $m \leq n$. We assume as our induction hypothesis that the translation is sound for such components and show that it also holds when $m \leq n+1$. Note that this assumption implies that all direct sub-**P**-schema of Ψ to which C links are translatable. We will refer to these direct sub-**P**-schema of Ψ, ignoring self referential links (these are handle as in the basecase), as $\Phi_1, \cdots, \Phi_k$, and each of them has a sequence of substitutions for each occurrence of the links in C, i.e. $\sigma_i^1, \cdots, \sigma_i^{r_i}$. We may now replace each of these links by an initial sequent $\mathcal{F}(es(\Phi_i))\sigma_i^j \vdash \mathcal{F}(es(\Phi_i))\sigma_i^j$. The result is the following end sequent of $C.3$:

$$\mathcal{F}(es(\Phi_1))\sigma_1^1,\cdots,\mathcal{F}(es(\Phi_1))\sigma_1^{r_1},\cdots,\mathcal{F}(es(\Phi_1))\sigma_k^{r_k},\mathcal{F}(es(C.3))\sigma_C^1,\cdots,\mathcal{F}(es(C.3))\sigma_C^{r_c}$$

$$\vdash \mathcal{F}(es(C.3))\{n \leftarrow n+1\}$$

We may now cut $\mathcal{F}(es(\Phi_i))\sigma_i^j$ using the translation of Φ_i and treat $\mathcal{F}(es(C.3)$ $)\sigma_C^{r_i}$ in a similar manor as in the basecase.

Now assuming that the theorem holds for $|\mathbf{P}| < n+1$ we can show that the theorem holds when $|\mathbf{P}| \leq n+1$. This result is quite a simple extension of the case when $|P| = 1$. Let us consider a component C s.t. all of its direct sub-**P**-schema have at most n passive parameters but C contain $n+1$. because Ψ is strict, there are no passive parameters below C which are not contained in C. Thus, the same construction can be applied again. □

Note that the translation defined in Lemma 19, along with Lemma 20, provides equivalence between strict **P**-schema and primitive recursive arithmetic (**PRA**) being that strict **P**-schema do not allow the expression of totality of any of the constructions.

Lemma 20. *Let Π be a strict mv**LKIE**-derivation of $\mathcal{S}$ containing α inductions of the form*

$$\frac{F_\beta(n, \mathbf{m}_1, \cdots, \mathbf{m}_\gamma), \Gamma_\beta \vdash \Delta_\beta, F_\beta(n', \mathbf{m}_1, \cdots, \mathbf{m}_\gamma)}{F_\beta(0, \mathbf{a}_1, \cdots, \mathbf{a}_\gamma), \Gamma_\beta \vdash \Delta_\beta, F_\beta(t, \mathbf{a}_1, \cdots, \mathbf{a}_\gamma)}\; mv\mathbf{IND}$$

where $1 \leq \beta \leq \alpha$, and if $\eta < \beta$ then the induction inference with conclusion $F_\beta(0, \mathbf{a}_1, \cdots, \mathbf{a}_\gamma), \Gamma_\beta \vdash \Delta_\beta, F_\beta(t, \mathbf{a}_1, \cdots, \mathbf{a}_\gamma)$ is above the induction inference *with conclusion $F_\eta(0, \mathbf{a}'_1, \cdots, \mathbf{a}'_{\gamma*}), \Gamma_\eta \vdash \Delta_\eta, F_\eta(t, \mathbf{a}'_1, \cdots, \mathbf{a}'_{\gamma*})$ in Π. Then there exists a strict* **P***-schema with end-sequent $\mathcal{S}$.*

Proof. Let T be the transformation taking an *mv***LKIE**-derivation φ to an *mv***LKS**-derivation by replacing the induction inferences $F_\eta(0, \mathbf{a}_1, \cdots, \mathbf{a}_\gamma)$, $\Gamma_\eta \vdash \Delta_\eta$, $F_\eta(t, \mathbf{a}_1, \cdots, \mathbf{a}'_\gamma)$, $\eta < \beta$, with a proof link $\psi_\eta(t, \mathbf{a}'_1, \cdots, \mathbf{a}'_{\gamma'})$. If the transformation reaches the induction inference β it replaces the $F_\beta(0, \mathbf{a}_1, \cdots, \mathbf{a}_\gamma)$, $\Gamma_\beta \vdash \Delta_\beta$, $F_\beta(t, \mathbf{a}_1, \cdots, \mathbf{a}_\gamma)$ with a proof link $\psi_\beta(t, \mathbf{m}_1, \cdots, \mathbf{m}_\gamma)$ and sequent of the proof link $F_\beta(0, \mathbf{m}_1, \cdots, \mathbf{m}_\gamma)$, $\Gamma_\beta \vdash \Delta_\beta, F_\beta(n, \mathbf{m}_1, \cdots, \mathbf{m}_\gamma)$. The instantiation is placed in the construction of the component for the predecessor of β. Such a transformation obviously constructs a strict **P**-schema from a strict *mv***LKIE**-derivation.

We will inductively construct a strict **P**-schema $\langle \mathbf{C}_1, \cdots, \mathbf{C}_\alpha \rangle$ where $\mathbf{C}_\beta = (\psi_\beta, \pi, \nu)$ has the end sequent $F_\beta(0, \mathbf{m}_1, \cdots, \mathbf{m}_\gamma), \Gamma_\beta \vdash \Delta_\beta, F_\beta(n, \mathbf{m}_1, \cdots, \mathbf{m}_\gamma)$ for some active parameter n. Assume that we have already constructed such proofs for $\mathbf{C}_{\beta+1}, \cdots, \mathbf{C}_\alpha$ and consider the induction inference with the following main sequent $F_\beta(0, \mathbf{a}_1, \cdots, \mathbf{a}_\gamma), \Gamma_\beta \vdash \Delta_\beta, F_\beta(t, \mathbf{a}_1, \cdots, \mathbf{a}_\gamma)$. Let ξ be the derivation above the induction. We set π to $F_\beta(0, \mathbf{m}_1, \cdots, \mathbf{m}_\gamma)$, $\Gamma_\beta \vdash \Delta_\beta$, $F_\beta(0, \mathbf{m}_1, \cdots, \mathbf{m}_\gamma)$ which by definition fulfills the requirements of links. Furthermore, let ν be the proof

$$\dfrac{\dfrac{\begin{array}{cc}\psi_\beta(n, \bar{\mathbf{m}}_\gamma) & T(\xi) \\ S(\psi_\beta(n, \bar{\mathbf{m}}_\gamma)) & S(T(\xi))\end{array}}{F_\beta(0, \bar{\mathbf{m}}_\gamma), \Gamma_\beta, \Gamma_\beta \vdash \Delta_\beta, \Delta_\beta, F_\beta(n', \bar{\mathbf{m}}_\gamma)}\ mv\mathbf{IND}}{F_\beta(0, \bar{\mathbf{m}}_\gamma), \Gamma_\beta \vdash \Delta_\beta, F_\beta(n', \bar{\mathbf{m}}_\gamma)}\ c*$$

where $S(\psi_\beta(n, \bar{\mathbf{m}}_\gamma)) \equiv F_\beta(0, \bar{\mathbf{m}}_\gamma), \Gamma_\beta \vdash \Delta_\beta, F_\beta(n, \bar{\mathbf{m}}_\gamma)$, $S(T(\xi)) \equiv F_\beta(n, \bar{\mathbf{m}}_\gamma)$, $\Gamma_\beta \vdash \Delta_\beta, F_\beta(n', \bar{\mathbf{m}}_\gamma)$, which also clearly satisfies the requirement on links. Summarizing, $\mathbf{C}_\beta$ is a component with end-sequent $F_\beta(0, \bar{\mathbf{m}}_\gamma), \Gamma_\beta \vdash \Delta_\beta, F_\beta(n, \bar{\mathbf{m}}_\gamma)$. Linkability and the partial ordering come for free from the construction of strict *mv***LKIE**-derivations. ☐

Notice that Lemma 20 does not put a restriction on the number of passive parameters in the end sequent, but limits the partial ordering of components to a total linear ordering. A simple corollary of Lemma 20 removes the restriction on the ordering of components, one can join sub-**P**-schema whose components are totally ordered into one whose components are not totally ordered. Proving the corollary requires the same induction argument over a the more complex order structure (the linkability ordering). We would have to join chains of components together using cuts.

Corollary 21. *Let Π be a strict mv**LKIE**-derivation of $\mathcal{S}$. Then there exists a strict* **P***-schema with end-sequent $\mathcal{S}$.*

Concerning strict *mv***LKIE**-derivations, notice that the need for passive, internal, and active parameters is no longer there. The three parameters aided the formalization of **P**-schema by removing mutual recursion and parameter instantiation, which are difficult to handle. Essentially, a reasonable class of strict

P-schema could not be constructed without the three types of parameters. But for strict *mv***LKIE**-derivations, the construction is obvious and enforced by the proof structure, thus, we can replace internal and active parameters by the corresponding constants and passive parameters. The resulting rule is essentially the induction rule of arithmetic. However, given the inclusion of the $\mathcal{E}$ rule, which allows arbitrarily complex equational theories, the language is at least a conservative extension of **PRA**if $\mathcal{E}$ is limited to functions provably total in **PRA**. We show that for a particular choice of equational theory and using the standard equational axioms, the $\mathcal{E}$ rule is admissible and thus strict *mv***LKIE**-derivations are precisely as expressive as **PRA** and by transitivity so is the strict **P**-schema formulation.

Furthermore, by dropping the strictness requirement, that is allowing computational sub **P**-schema, It follows that the **P**-schema formulation is provability equivalent to **PA**.

4 P-schema, PRA, and PA

We will consider the **P**-schema formulation over the following equational theory

$$\mathcal{E}_{\mathrm{PA}} = \left\{ \widehat{a}(s(n),\beta)=s(\widehat{a}(n,\beta)) \ ; \ \widehat{a}(0,\beta)=\beta \widehat{m}(s(n),\beta)=\widehat{a}(\widehat{m}(n,\beta),\beta) \ ; \ \widehat{m}(0,\beta)=0 \right\}.$$

Furthermore we we extend the **LK**-calculus by the following initial sequents of arithmetic and equational reasoning over the ω sort:

$$\begin{array}{lll} \vdash t = t \quad s(0) = 0 \vdash & & s_1 = t_1, \cdots s_n = t_n \vdash f(s_1, \cdots s_n) = f(t_1, \cdots t_n) \\ s(x) = s(y) \vdash x = y & & s_1 = t_1, \cdots s_n = t_n, P(s_1, \cdots s_n) \vdash P(t_1, \cdots t_n) \end{array}$$

More information concerning equational reasoning and the axioms of arithmetic may be found in [22]. We will refer to these axioms as $\mathrm{Ax}_{\mathrm{PA}}$. The **LK**-calculus extended by these initial sequents, the definition of addition and multiplication as well as the induction inference (as discussed in [22]) will be referred to as the **PA**-calculus (or **PRA**-calculus when we are referring to primitive recursive arithmetic alone).

Lemma 22. *Let Π be a strict mv**LKIE**-proof using Ax_{PA} and $\mathcal{E}_{\mathrm{PA}}$. Then there exists a strict mv**LKIE**-proof Π' without the $\mathcal{E}$ inference rule (Π' is $\mathcal{E}$-free) ending with $es(\Pi)$.*

Proof. The rewrite rules of $\mathcal{E}_{\mathrm{PA}}$ are precisely the axioms of $\mathrm{Ax}_{\mathrm{PA}}$ for addition and multiplication. Thus, from those axioms and the initial sequents introduced above anything provable by the $\mathcal{E}$ inference rule can be proven using the above mentioned axioms and atomic cuts.

Now that we have $\mathcal{E}$-free strict *mv***LKIE**-proofs we can consider translation to the **PA**-calculus without quantification. As the end sequent of a strict *mv***LKIE**-proof only has passive parameters. We can push the passive parameters up the proof tree and replace each active parameter by a fresh passive parameter. Thus, the resulting proof only contains passive parameters and constants and is a proof in the **PRA**-calculus.

Theorem 23. *There exists a $\mathcal{E}$-free strict mv**LKIE**-proof of a sequent S iff there exists a **PRA**-calculus proof of S.*

Proof. Let us refer to the $\mathcal{E}$-free strict *mv***LKIE**-proof of the sequent S as Φ. Note that on any branch starting at the root, the sequents of the branch are internal parameter free until we reach a *mv***LKIE** induction rule. Thus if Φ does not contain any *mv***LKIE**induction rules it is a **PRA**-calculus proof of S. Let us assume if for any proof Φ s.t. every branch starting from the root contains at most n *mv***LKIE** induction inferences than we can transform the proof in a **PRA**-calculus proof of S. We now show that the same is possible when there is a single branch which contain $n+1$ *mv***LKIE** induction inferences. Let us consider the lowest *mv***LKIE** induction inference occurring on the branch with $n+1$ inferences.

$$\dfrac{\begin{array}{c}(\varphi)\\ F(n), \Gamma \vdash \Delta, F(n+1)\end{array}}{F(0)\sigma, \Gamma\sigma \vdash \Delta\sigma, F(t)\sigma}\ mv\mathbf{IND}$$

We can apply the substitution to the auxiliary sequent and all sequents of φ which are not auxiliary sequents of an *mv***LKIE** induction inference. This application of substitution transforms the *mv***IND** inference in the standard induction inference and by the induction hypothesis the result is a **PRA**-calculus proof of S. □

Example 24. The following proof of commutativity is the result of applying the translation from $\mathcal{E}$-free strict *mv***LKIE**-proofs to **PA**-calculus proofs. Note that the proof below was first introduced in Example 11.

$$\dfrac{\dfrac{\begin{array}{c}a(\boldsymbol{\mu},1)=a(1,\boldsymbol{\mu})\vdash\\ a(s(\boldsymbol{\mu}),1)=a(1,s(\boldsymbol{\mu}))\end{array}}{\begin{array}{c}a(0,1)=a(1,0)\vdash\\ a(\boldsymbol{\alpha},1)=a(1,\boldsymbol{\alpha})\end{array}}\ \mathbf{IND}\quad\vdots \qquad \dfrac{\begin{array}{c}\vdots\\ a(\boldsymbol{\nu},a(1,\boldsymbol{\gamma}))=a(a(\boldsymbol{\nu},1),\boldsymbol{\gamma})\vdash\\ a(s(\boldsymbol{\nu}),a(1,\boldsymbol{\gamma}))=a(a(s(\boldsymbol{\nu}),1),\boldsymbol{\gamma})\end{array}}{\begin{array}{c}a(0,a(1,\boldsymbol{\gamma}))=a(a(0,1),\boldsymbol{\gamma})\vdash\\ a(\boldsymbol{\alpha},a(1,\boldsymbol{\gamma}))=a(a(\boldsymbol{\alpha},1),\boldsymbol{\gamma})\end{array}}\ \mathbf{IND}\quad \begin{array}{c}\vdots\\ \vdash a(\boldsymbol{\alpha},a(1,\boldsymbol{\gamma}))=a(a(\boldsymbol{\alpha},1),\boldsymbol{\gamma})\end{array}}{\begin{array}{c}a(\boldsymbol{\alpha},\boldsymbol{\gamma})=a(\boldsymbol{\gamma},\boldsymbol{\alpha})\vdash\\ a(\boldsymbol{\alpha},a(1,\boldsymbol{\gamma}))=a(s(\boldsymbol{\gamma}),\boldsymbol{\alpha})\\ (1)\end{array}}\ cut$$

$$\dfrac{\dfrac{(1)\qquad \begin{array}{c}a(\boldsymbol{\alpha},a(1,\boldsymbol{\gamma}))=a(s(\boldsymbol{\gamma}),\boldsymbol{\alpha}),\\ \boldsymbol{\alpha}=\boldsymbol{\alpha}, s(\boldsymbol{\gamma})=s(\boldsymbol{\gamma}),\\ a(1,\boldsymbol{\gamma})=s(a(0,\boldsymbol{\gamma}))\vdash\\ a(\boldsymbol{\alpha},s(a(0,\boldsymbol{\gamma})))=a(s(\boldsymbol{\gamma}),\boldsymbol{\alpha})\end{array}}{\vdots}\ cut \qquad \begin{array}{c}\vdots\\ a(\boldsymbol{\alpha},s(a(0,\boldsymbol{\gamma})))=a(s(\boldsymbol{\gamma}),\boldsymbol{\alpha}),\\ \boldsymbol{\alpha}=\boldsymbol{\alpha}, s(\boldsymbol{\gamma})=s(\boldsymbol{\gamma}),\\ s(a(0,\boldsymbol{\gamma}))=s(\boldsymbol{\gamma})\vdash\\ a(\boldsymbol{\alpha},s(\boldsymbol{\gamma}))=a(s(\boldsymbol{\gamma}),\boldsymbol{\alpha})\end{array}}{\vdots}\ cut$$

$$\dfrac{\dfrac{\vdots}{\begin{array}{c}a(\boldsymbol{\alpha},\boldsymbol{\gamma})=a(\boldsymbol{\gamma},\boldsymbol{\alpha})\vdash\\ a(\boldsymbol{\alpha},s(\boldsymbol{\gamma}))=a(s(\boldsymbol{\gamma}),\boldsymbol{\alpha})\end{array}}}{\begin{array}{c}a(\boldsymbol{\alpha},0)=a(0,\boldsymbol{\alpha})\vdash\\ a(\boldsymbol{\alpha},\boldsymbol{\beta})=a(\boldsymbol{\beta},\boldsymbol{\alpha})\end{array}}\ \mathbf{IND}$$

To get from **PRA** to **PA** we need to remove the requirement of only considering strict *mv***LKIE**-proofs. In terms of **P**-schema, this would mean allowing computational sub-**P**-schema. We can build a *mv***LKIE**-proof χ containing a sub-derivation ψ which is a strict *mv***LKIE**-proof by allowing strong quantification on the passive parameters of ψ in χ. The problem is that doing so can possibly destroy the translation of Section 3. To show this is not possible we just have to consider translation of χ in parts, first we translate ψ and then we translate χ without ψ, that is replacing ψ with a theory axiom during translation. Once we finish the translation of both parts we glue them back together to get a translation of the original proof χ. This results in the following theorems:

Theorem 25. *A* **P***-schema* Ψ *with* $es(\Psi) = \mathcal{S}$ *exists iff a mv**LKIE**-derivation of* $\mathcal{S}$ *exists.*

Proof. We can convert Ψ into an *mv***LKIE**-derivation by structural induction over the number of passive parameters not associated with computational sub-**P**-schema. First we consider strict **P**-schema (Theorem 23). As the induction hypothesis we assume that the theorem holds for the first n computational sub-**P**-schema of Ψ, then we show it for $n + 1$.

Finally, we can extend the results of this section to **PA**.

Theorem 26. *There is an* $\mathcal{E}$*-free mv**LKIE**-derivation of a sequent* S *iff there is a **PA**-calculus proof of* S*.*

Proof. Note that the **PA** induction rule is a special case of the *mv***LKIE** induction rule. Thus, forward translation can be handled using the above results and backwards translation, from **PA** to an *mv***LKIE**-derivation, trivial.

5 Conclusion

In this paper we generalized the proof schemata formalism of [15] to a much larger fragment of arithmetic. We refer to this generalization as **P**-schema and prove that a subset of **P**-schema, which we refer to as strict **P**-schema, is equivalent to **PRA** by providing a translation between strict **P**-schema and a standard formulation of primitive recursive arithmetic using the **LK**-calculus. Furthermore, we show that removing this restriction results in a formalism equivalent **PA** formalized as a theory extension of the **LK**-calculus.

Concerning future work, we plan to address the issues raised in the introduction concerning computational proof analysis in the presence of induction and the extraction of Π_1 computational content. This work provides evidence that the proof schema formalism introduced in earlier work on computational proof analysis [15] is as expressive the standard sequent calculus for Peano arithmetic. We plan to investigate the use of the proof transformation methods introduce in [15] on **P**-schema as well as apply the resulting proof transformation method.

Acknowledgments: We would like to thank Michael Lettmann, Alexander Leitsch and Matthias Baaz for contributing to the polishing of this work.

References

1. Jeremy Avigad and Henry Towsner. Functional interpretation and inductive definitions. *J. Symb. Log.*, 74(4):1100–1120, 2009.
2. 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, August 2008.
3. Matthias Baaz and Alexander Leitsch. Cut-elimination and redundancy-elimination by resolution. *Journal of Symbolic Computation*, 29:149–176, 2000.
4. Ulrich Berger, Wilfried Buchholz, and Helmut Schwichtenberg. Refined program extraction from classical proofs. *Annals of Pure and Applied Logic*, 114(1):3 – 25, 2002. Troelstra Festschrift.
5. James Brotherston. Cyclic proofs for first-order logic with inductive definitions. In *Tableaux'05*, volume 3702 of *Lecture Notes in Comp. Sci.*, pages 78–92. 2005.
6. James Brotherston and Alex Simpson. Sequent calculi for induction and infinite descent. *Journal of Logic and Computation*, 21(6):1177–1216, 2010.
7. Wilfried Buchholz. Notation systems for infinitary derivations. *Archive for Mathematical Logic*, 30(5-6):277–296, 1991.
8. David M. Cerna and Michael Lettmann. Integrating a global induction mechanism into a sequent calculus. In *TABLEAUX'17*, Lecture Notes in Comp. Sci., Sept. 2017.
9. David M. Cerna and Michael Lettmann. Towards a clausal analysis of proof schemata. In *SYNASC'17*, IEEE Xplorer. IEEE, Sept. 2017.
10. Harvey Friedman. Classically and intuitionistically provably recursive functions. In Gert H. Müller and Dana S. Scott, editors, *Higher Set Theory*, pages 21–27, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg.
11. Gerhard Gentzen. Fusion of several complete inductions. In M.E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, volume 55 of *Studies in Logic and the Foundations of Mathematics*, pages 309 – 311. Elsevier, 1969.
12. Kurt Gödel. Über eine bisher noch nicht bentzte erweiterung des finiten standpunktes. *Dialectica*, 12(3–4):280–287, 1958.
13. Stefan Hetzl and Tin Lok Wong. Restricted notions of provability by induction. abs/1704.01930, 2017.
14. Jeroen Ketema, Jan Willem Klop, and Vincent van Oostrom. Vicious circles in orthogonal term rewriting systems. *Electr. Notes Theor. Comput. Sci.*, 124(2):65–77, 2005.
15. Alexander Leitsch, Nicolas Peltier, and Daniel Weller. CERES for first-order schemata. *J. Log. Comput.*, 27(7):1897–1954, 2017.
16. Raymond Mcdowell and Dale Miller. Cut-elimination for a logic with definitions and induction. *Theoretical Computer Science*, 232, 1997.
17. Reuben N. S. Rowe and James Brotherston. Realizability in cyclic proof: Extracting ordering information for infinite descent. In *TABLEAUX 2017 Proceedings*, pages 295–310, 2017.
18. Helmut Schwichtenberg. Realizability interpretation of proofs in constructive analysis. *Theory Comput. Syst.*, 43(3-4):583–602, 2008.
19. J. R. Shoenfield. On a restricted ω-rule. *Bulletin de l'Acadimie Polonaise des Sciences*, 7:405–407, 1959.
20. Harold Simmons. The realm of primitive recursion. *Archive for Mathematical Logic*, 27(2):177–188, Sep 1988.

21. Alex Simpson. Cyclic arithmetic is equivalent to peano arithmetic. In *FOSSACS 2017, Proceedings*, pages 283–300, 2017.
22. Gaisi Takeuti. *Proof Theory*, volume 81 of *Studies in logic and the foundations of mathematics*. American Elsevier Pub., 1975.