EPiC Series in Computing
Volume 100, 2024, Pages 295–310
Proceedings of 25th Conference on Logic for Programming, Artificial Intelligence and Reasoning

Herbrand's Theorem in Inductive Proofs

Alexander Leitsch[1] and Anela Lolić[2]*

[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

Abstract

An inductive proof can be represented as a proof schema, i.e. as a parameterized sequence of proofs defined in a primitive recursive way. A corresponding cut-elimination method, called schematic CERES, can be used to analyze these proofs, and to extract their (schematic) Herbrand sequents, even though Herbrand's theorem in general does not hold for proofs with induction inferences. This work focuses on the most crucial part of the schematic cut-elimination method, which is to construct a refutation of a schematic formula that represents the cut-structure of the original proof schema. Moreover, we show that this new formalism allows the extraction of a structure from the refutation schema, called a Herbrand schema, which represents its Herbrand sequent.

1 Introduction

Herbrand's theorem [8] is one of the most important results of mathematical logic. It expresses the fact that in a formal cut-free proof of a prenex form propositional and quantifier inferences can be separated. In the formalism of sequent calculus this means that a so-called Herbrand sequent can be extracted from a proof, where the propositional inferences operate above the sequent, and the quantifier inferences below. In automated theorem proving Herbrand's theorem is used as a tool to prove completeness of refinements of resolution. Moreover the theorem can yield a compact representations of proofs by abstracting away the propositional inferences. Such compact representations play also a major role in computational proof analysis: formal proofs obtained by cut-elimination from formalized mathematical proofs are typically very long covering up the mathematical content of the proofs. Experiments with the system CERES (cut-elimination by resolution, see [5] and [6]) have revealed that Herbrand forms display the main mathematical arguments of a proof in a natural way [9].

As most interesting mathematical proofs contain applications of mathematical induction, an extension of CERES to the analysis of inductive proofs was of major importance to turn

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

the method into a practically useful tool for (interactive) proof analysis. A first thorough analysis of an inductive (schematic) CERES method can be found in [11]. 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 schema* was developed, which is essentially an extension of Herbrand's theorem from single proofs to a (recursively defined) infinite sequence of proofs. This definition of a Herbrand schema represented the first step to extend Herbrand's theorem to inductive proofs (note that proofs using the induction rule do not admit a construction of Herbrand sequents). In [12] the approach in [11] was extended to arbitrary many induction parameters thus considerably increasing the strength of the method.

The core of CERES consists of the construction of a resolution refutation schema of a formula schema which represents the derivations of the cut formulas in the original proof schema. This refutation schema can then be combined with a so-called projection schema, which can be obtained from the original proof schema by omitting all cut-inferences. It was shown in [12] that for the construction of the Herbrand schema of the original proof schema, the Herbrand schemata from the refutation schema and the projection can be combined. Hereby the most complex task consists in the computation of the Herbrand schema of the refutation schema, which justifies the investigation of Herbrand schemata in refutation schemata on its own. Indeed, the Herbrand schema of a refutation schema R may reveal crucial mathematical information contained in R. As an example the analysis of Fürstenberg's proof of the infinitude of primes in [4] (still not formalized but carried out on the mathematical meta-level) could be mentioned: here the schematic Herbrand instances represent Euclid's construction of primes.

In [7] the calculus for refuting resolution proof schemata as defined in [12] was substantially extended by the use of point transition systems. The expressivity of the new calculus was demonstrated on a formula schema which could not be refuted by earlier methods. However, the construction of Herbrand schemata was not possible in [7], mainly due to a missing formalism for handling schematic substitutions and schematic unifications.

In this paper we present a novel calculus for refuting schematic formulas, and introduce a notion of refutation schema that considerably simplifies previous notions of proof as schema. Moreover, we demonstrate that this new framework allows the construction of a Herbrand schema from a resolution refutation schema, thus paving the way for proof analysis methods in presence of induction inferences.

2 The Resolution Calculus RPL_0

The basis for the schematic refutational calculus is the calculus RPL_0 for quantifier-free formulas, as introduced in [7]. This calculus combines dynamic normalization rules à la Andrews [1] with the resolution rule, but in contrast to [1] does not restrict the resolution rule to atomic formulas.

The main motivation of the calculus RPL_0 is that it can be extended to a schematic setting in a straightforward way, and that it is particularly suited for the extraction of Herbrand substitutions in the form of a Herbrand schemata of the schematic refutations.

The set of quantifier-free formulas in predicate logic will be denoted as PL_0, and for simplicity we omit $\to$, but can represent it by $\neg$ and $\vee$ in the usual way. In this setting, as sequents we consider objects of the form $\Gamma \vdash \Delta$, where Γ and Δ are multisets of formulas in PL_0.

Definition 1 (RPL_0)**.** *The axioms of* RPL_0 *are sequents* $\vdash F$ *for* $F \in \mathrm{PL}_0$.

The rules are elimination rules for the connectives and the resolution rule.

$$\frac{\Gamma \vdash \Delta, A \wedge B}{\Gamma \vdash \Delta, A} \wedge\colon r_1 \quad \frac{\Gamma \vdash \Delta, A \wedge B}{\Gamma \vdash \Delta, B} \wedge\colon r_2 \quad \frac{A \wedge B, \Gamma \vdash \Delta}{A, B, \Gamma \vdash \Delta} \wedge\colon l$$

$$\frac{\Gamma \vdash \Delta, A \vee B}{\Gamma \vdash \Delta, A, B} \vee\colon r \quad \frac{A \vee B, \Gamma \vdash \Delta}{A, \Gamma \vdash \Delta} \vee\colon l_1 \quad \frac{A \vee B, \Gamma \vdash \Delta}{B, \Gamma \vdash \Delta} \vee\colon l_2$$

$$\frac{\Gamma \vdash \Delta, \neg A}{A, \Gamma \vdash \Delta} \neg\colon r \quad \frac{\neg A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, A} \neg\colon l$$

The resolution rule, where σ is an m.g.u. of $\{A_1, \ldots, A_k, B_1, \ldots, B_l\}$ and $V(\{A_1, \ldots, A_k\}) \cap V(\{B_1, \ldots, B_l\}) = \emptyset$ is

$$\frac{\Gamma \vdash \Delta, A_1, \ldots, A_k \quad B_1, \ldots, B_m, \Pi \vdash \Lambda}{\Gamma\sigma, \Pi\sigma \vdash \Delta\sigma, \Lambda\sigma} \; res$$

An RPL_0*-derivation is a tree formed from axioms* $\vdash F\theta$ *by application of the rules above where* F *is a formula in* PL_0 *and* θ *is a variable renaming. Therefore, an* RPL_0*-derivation is defined relative to* F*. An* RPL_0*-derivation is called* regular *if any two different axioms are variable-disjoint; so if there are* k *different axioms in a regular* RPL_0*-derivation they are of the form* $\vdash F\theta_1, \ldots, \vdash F\theta_k$ *and* $V(F\theta_i) \cap V(F\theta_j) = \emptyset$ *for* $i \neq j$*. An* RPL_0-refutation *of* F *is a* RPL_0*-derivation of* $\vdash$ *with axioms of the form* $\vdash F\theta_i$*.*

In [7] RPL_0 is shown to be sound and refutationally complete.

In general, several resolution rules may occur in a RPL_0-derivation, and hence several most general unifiers σ_i need to be applied. In a regular derivation, a total unifier (or total m.g.u.) can be obtained by considering the most general simultaneous unifier of the unification problems given by the atoms in the premises of all resolution rules.

Definition 2 (simultaneous unifier)**.** *Let* $W = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$*, where the* $\mathcal{A}_i$ *are nonempty sets of atoms for* $i = 1, \ldots, n$*. A substitution* σ *is called a simultaneous unifier of* W *if* σ *unifies all* $\mathcal{A}_i$*.* σ *is called a most general simultaneous unifier of* W *if* σ *is a simultaneous unifier of* W *and* $\sigma \leq_s \sigma'$ *for all simultaneous unifiers* σ' *of* W*.*

Definition 3 (total m.g.u.)**.** *Let* ρ *be a regular* RPL_0*-derivation containing* n *resolution inferences. The unification problem of* ρ *is defined as* $W = (\mathcal{A}_1, \ldots, \mathcal{A}_n)$*, where* $\mathcal{A}_i$ *(*$i \in \{1, \ldots n\}$*) is the set* $\{A^i_1, \ldots, A^i_{k_i}, B^i_1, \ldots, B^i_{m_i}\}$ *in the resolution inferences*

$$\frac{\Gamma \vdash \Delta, A^i_1, \ldots, A^i_{k_i} \quad B^i_1, \ldots, B^i_{m_i}, \Pi \vdash \Lambda}{\Gamma\sigma_i, \Pi\sigma_i \vdash \Delta\sigma_i, \Lambda\sigma_i} \; res$$

in ρ*. If* σ *is a most general simultaneous unifier of* W*,* σ *is called a total m.g.u. of* ρ*.*

Note that, after application of a total m.g.u. to a regular RPL_0-derivation the resolution rules become cut rules.

3 Schematic Language

In this work we will use the many-sorted version of classical first-order logic, as introduced in [7], and define schemata based on primitive recursion. Due to space limitations, we refer the interested reader to [7] and to [10] 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 $s(N) = s(N') \to N = N'$. 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 [7]).

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 [10], 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 has arity α and $t_1, \dots, t_\alpha \in \mathcal{T}^\iota$ then $f(t_1, \dots, t_\alpha) \in \mathcal{T}^\iota$.
- 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 [7] and [10]. 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 $t_1, \dots, t_\alpha \in T^\iota$ and $P \in \mathcal{P}$ s.t. P has arity $\alpha \geq 0$, then $P(\overrightarrow{t_\alpha}) \in \mathcal{T}_V^o$.
- 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 $\mathcal{T}_0^o$.

For defining formula schemata we extend the concept of individual variables to so-called *global* variables: Variables which take numeric arguments, i.e. $X(\overrightarrow{t}_\alpha)$, where $\overrightarrow{t}_\alpha \in T^\omega$ for $\alpha \geq 0$ (note that α is fixed and finite). The set of all global variables will be denoted by V^G, and terms of the form $X(\overrightarrow{t}_\alpha)$ will be referred to as *V-terms over* X. The set of V-terms whose arguments are numerals (from *Num*) will be denoted by V^ι. Such terms are referred to as *individual variables*. We will often denote the set of individual variables contained in some object $\mathbf{T}$ by $V^\iota(\mathbf{T})$, e.g. a substitution, an ι term, a set of ι terms, etc. Note that global variables could be interpreted as second-order variables, but they are never quantified nor are they subject to second-order substitutions - they are in some sense *passive* second-order variables.

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 $t_1, \cdots, t_\alpha \in \mathcal{T}^o$, $\hat{P} \in \hat{\mathcal{P}}$, $\overrightarrow{X_\beta} \in V^G$, and $\overrightarrow{n_{\alpha+1}} \in \mathcal{N}$ s.t. $\hat{P}$ has arity $\alpha + \beta + 1$ for $\alpha, \beta \geq 0$, then $\hat{P}(\overrightarrow{X_\beta}, \overrightarrow{n_{\alpha+1}}) \in \mathcal{T}^o$.
- 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 4 (defining equations)**.** *Let $\hat{p} \in \hat{\sigma}_o$. We define a set $D(\hat{p})$ consisting of two equations:*

$$\hat{p}(\overrightarrow{X}_\alpha, \overrightarrow{n}_\beta, \bar{0}) = \hat{p}_B, \quad \hat{p}(\overrightarrow{X}_\alpha, \overrightarrow{n}_\beta, s(m)) = \hat{p}_S\{\xi \leftarrow \hat{p}(\overrightarrow{X}_\alpha, \overrightarrow{n}_\beta, m)\}, \text{ where}$$

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 $\hat{p}$ is non-minimal: $\hat{p}_B, \hat{p}_S \in \mathcal{T}^o$ where $\hat{p}_B, \hat{p}_S$ may contain only defined function 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 global variables occurring in $\hat{p}_B$ and $\hat{p}_S$ are $\overrightarrow{X}_\alpha$. 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. Furthermore, the evaluation of equations can be nonterminating if we omit condition 2 of the definition above. However, in this work the equations can be oriented to terminating and confluent rewrite systems and thus termination of the evaluation procedure is easily verified [7].

Definition 5 (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 $\mathcal{T}_0^o$.

The last point we would like to make concerning terms $\mathcal{T}^o$ is that we designed the language to finitely express *infinite sequences of quantifier-free first-order formulas*. In particular, we are interested in infinite sequences of unsatisfiable formulas whose refutations are finitely describable using the resolution calculus introduced later in this paper.

Definition 6 (unsatisfiable schemata)**.** *Let $F \in \mathcal{T}^o$. Then F is called* unsatisfiable *if for all $\sigma \in \mathcal{S}$ the formula $\sigma(F)\downarrow$ is unsatisfiable (see [7], page 607, Definition 7).*

Example 3. *Let $a \in \Sigma_\iota$, $P \in \Sigma_o$, $\hat{f}$ as in Example 2, $\hat{p}, \hat{q} \in \hat{\Sigma}_o$ s.t. $\hat{p}$ is smaller than $\hat{Q}$. We consider the theory $(\{\hat{p}, \hat{q}, \hat{f}\}, \hat{q}, \{D(\hat{p}), D(\hat{q}), D(\hat{f})\})$. The defining equations for $\hat{p}$ and $\hat{q}$ are:*

$\hat{p}(X, \bar{0}) = \neg P(X(\bar{0}), \hat{f}(a, 0))$, $\hat{p}(X, s(n)) = \hat{p}(X, n) \vee \neg P(X(s(n)), \hat{f}(a, s(n)))$,

$\hat{q}(X, Y, n, \bar{0}) = P(\hat{f}(Y(\bar{0}), \bar{0}), Y(\bar{1})) \wedge \hat{p}(X, n)$ *and*

$\hat{q}(X, Y, n, s(m)) = P(\hat{f}(Y(\bar{0}), s(m)), Y(\bar{1})) \wedge \hat{p}(X, n)$.

It is easy to see that the schema $\hat{q}(X, Y, n, m)$ *is unsatisfiable. Let us consider* $\sigma(\hat{q}(X, Y, n, m))\downarrow$ *for* σ *with* $\sigma(m) = \bar{2}, \sigma(n) = \bar{3}$*:*

$$
\begin{aligned}
&\sigma(\hat{q}(X, Y, n, m))\downarrow = \hat{q}(X, Y, \bar{3}, \bar{2}) = P(\hat{f}(Y(\bar{0}), \bar{2}), Y(\bar{1})) \wedge \hat{p}(X, 3)\downarrow = \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots = \\
&P(f(f(Y(\bar{0}))), Y(\bar{1})) \wedge \\
&(\neg P(X(\bar{0}), a) \vee \neg P(X(\bar{1}), f(a)) \vee \neg P(X(\bar{2}), f(f(a))) \vee \neg P(X(\bar{3}), f(f(f(a))))).
\end{aligned}
$$

Note that for $\sigma(n) = \bar{\alpha}$ *the number of different variables in* $\sigma(\hat{q}(X, Y, n, m))\downarrow$ *is* $\alpha + 2$*, so the number of variables increases with the increase of* $\sigma(n)$*.*

4 Refutation Schemata

In this section we will extend RPL_0 by rules handling schematic formula definitions. In inductive proofs the use of lemmas is vital, i.e. an ordinary refutational calculus, which has just a weak capacity of lemma generation, may fail to derive the desired invariant. Therefore, we will add introduction rules for the connectives, giving us the potential to derive more complex formulas. Furthermore, we have to ensure that the formulas on which the resolution rule is applied have pairwise disjoint variables. We need a corresponding concept of disjointness for the schematic case.

Definition 7. *Let* A, B *be finite sets of schematic variables.* A *and* B *are called essentially disjoint if for all* $\sigma \in \mathcal{S}$ $\sigma(A)\downarrow \cap \sigma(B)\downarrow = \emptyset$*.*

Definition 8 (RPL_0^{Ψ})**.** *Let* $\Psi \colon (\mathcal{P}, \hat{q}, D(\mathcal{P}))$ *be a theory, then for all schematic predicate symbols* $\hat{p} \in \mathcal{P}$ *for* $D(\hat{p})(\vec{Y}, \vec{n}, 0) = \hat{p}_B$*, and* $D(\hat{p})(\vec{Y}, \vec{n}, s(m)) = \hat{p}_S\{\xi \leftarrow \hat{p}(\vec{Y}, \vec{n}, p(m))\}$*, we define the elimination of defined symbols*

$$
\frac{\Gamma \vdash \Delta, \hat{p}(\vec{Y}, \vec{n}, 0)}{\Gamma \vdash \Delta, \hat{p}_B}\ B\hat{p}r \qquad \frac{\Gamma \vdash \Delta, \hat{p}(\vec{Y}, \vec{n}, s(m))}{\Gamma \vdash \Delta, \hat{p}_S\{\xi \leftarrow \hat{p}(\vec{Y}, \vec{n}, m)\}}\ S\hat{p}r
$$

$$
\frac{\hat{p}(\vec{Y}, \vec{n}, 0), \Gamma \vdash \Delta}{\hat{p}_B, \Gamma \vdash \Delta}\ B\hat{p}l \qquad \frac{\hat{p}(\vec{Y}, \vec{n}, s(m)), \Gamma \vdash \Delta}{\hat{p}_S\{\xi \leftarrow \hat{p}(\vec{Y}, \vec{n}, m)\}, \Gamma \vdash \Delta}\ S\hat{p}l
$$

and the introduction of defined symbols

$$
\frac{\Gamma \vdash \Delta, \hat{p}_B}{\Gamma \vdash \Delta, \hat{p}(\vec{Y}, \vec{n}, 0)}\ B\hat{p}r^{+} \qquad \frac{\Gamma \vdash \Delta, \hat{p}_S\{\xi \leftarrow \hat{p}(\vec{Y}, \vec{n}, m)\}}{\Gamma \vdash \Delta, \hat{p}(\vec{Y}, \vec{n}, s(m))}\ S\hat{p}r^{+}
$$

$$
\frac{\hat{p}_B, \Gamma \vdash \Delta}{\hat{p}(\vec{Y}, \vec{n}, 0), \Gamma \vdash \Delta}\ B\hat{p}l^{+} \qquad \frac{\hat{p}_S\{\xi \leftarrow \hat{p}(\vec{Y}, \vec{n}, m)\}, \Gamma \vdash \Delta}{\hat{p}(\vec{Y}, \vec{n}, s(m)), \Gamma \vdash \Delta}\ S\hat{P}l^{+}
$$

As the theory Ψ *contains the theory for schematic terms* T' *and, due to the defining equations for schematic term symbols,* T' *is an equational theory we also need inference rules* within *schematic formulas. Let* t *be a term in* T' *of the form* (1)*:* $\hat{s}(s_1, \ldots, s_i, t_1, \ldots, t_{j-1}, s(w))$ *or of the form* (2)*:* $\hat{s}(s_1, \ldots, s_i, t_1, \ldots, t_{j-1}, \bar{0})$ *where* $D(\hat{s}) =$

$$
\begin{aligned}
&\{\hat{s}(x_1, \ldots, x_i, n_1, \ldots, n_{j-1}, s(n_j)) = \hat{s}_S\{z \leftarrow \hat{s}(x_1, \ldots, x_i, n_1, \ldots, n_{j-1}, n_j)\}, \\
&\hat{s}(x_1, \ldots, x_i, n_1, \ldots, n_{j-1}, \bar{0}) = \hat{s}_B\}
\end{aligned}
$$

Then, in case (1) we define $t \sim_{T'} t'$ *for*

$$t' = \hat{s}_S\theta\{z \leftarrow \hat{s}(s_1, \ldots, s_i, t_1, \ldots, t_{j-1}, w)\},$$

where $\theta = \{x_1 \leftarrow s_1, \ldots, x_i \leftarrow s_i, n_1 \leftarrow t_1, \ldots, n_{j-1} \leftarrow t_{j-1}, n_j \leftarrow w\}$.
In case (2) we define $t \sim_{T'} t'$ *for*

$$t' = \hat{s}_B\{x_1 \leftarrow s_1, \ldots, x_i \leftarrow s_i, n_1 \leftarrow t_1, \ldots, n_{j-1} \leftarrow t_{j-1}\}.$$

We also define $\sim_{T'}$ *as the reflexive and symmetric closure of the relation defined above. Now let* F *be a schematic formula occurring in* Ψ *such that* t *occurs at position* λ *in* F *and* $t \sim_{T'} t'$. *Then* $F[t']_\lambda$ *is called a* T'-variant *of* F. *Now let* $S\colon F_1, \ldots, F_i \vdash G_1, \ldots, G_j$ *be a sequent, the* F'_α *be* T'*-variants of* F_α *and the* G'_β *be* T'*-variants of* G_β. *Then we define the inference*

$$\frac{F_1, \ldots, F_i \vdash G_1, \ldots, G_j}{F'_1, \ldots, F'_i \vdash G'_1, \ldots, G'_j}\ T'$$

We also adapt the resolution rule to the schematic case:
Let $T^\iota_V(\{A_1, \ldots, A_\alpha\}), T^\iota_V(\{B_1, \ldots, B_\beta\})$ *be essentially disjoint sets of schematic variables and* Θ *be an s-unifier of* $\{A_1, \ldots, A_\alpha, B_1, \ldots, B_\beta\}$. *Then the resolution rule is defined as*

$$\frac{\Gamma \vdash \Delta, A_1, \ldots, A_\alpha \quad B_1, \ldots, B_\beta, \Pi \vdash \Lambda}{\Gamma\Theta, \Pi\Theta \vdash \Delta\Theta, \Lambda\Theta}\ res\{\Theta\}$$

In the definition above we use the notion of *s-substitution*, which stands for schematic substitution (for a formal definition see [10], page 7, Definition 3.1). Note that for two term schemata to be unifiable, they have to be unifiable for all possible parameter assignments. A unification algorithm is given in [10], page 27 (note that unification is not only undecidable, but necessarily incomplete). Here the use of global variables plays a vital role. Although there are unifiable term schemata that are defined without global variables, allowing this kind of indexed variables in the construction of term schemata simplifies the formalism, as described in [7]. The refutational completeness of RPL_0^Ψ is not an issue as already RPL_0 is refutationally complete for PL_0 formulas [3,11]. Note that this is not the case anymore if parameters occur in formulas. Indeed, due to the usual theoretical limitations, the logic is not semi-decidable for schematic formulas [2]. RPL_0^Ψ is sound.

Proposition 1. *Let the sequent* S *be derivable in* RPL_0^Ψ *for* $\Psi\colon (\mathcal{P}, \hat{q}, D(\mathcal{P}))$. *Then* $D(\mathcal{P}) \models S$.

Proof. The introduction and elimination rules for defined predicate symbols are sound (we have to consider the equations $D(\mathcal{P})$), as are the T' rules; also the resolution rule (involving s-unification) is sound. □

Before giving a formal definition of refutation schemata let us have a look at the following example. The formula schema below is a characteristic formula schema of a proof schema Φ defined in [12]; thus its refutation is a key step in the analysis of Φ via the schematic CERES method.

Example 4. *We define a refutation schema for the theory*

$$\Psi : ((\hat{q}, \mathcal{P}, D(\mathcal{P})), \vec{Y}, <), T', T''),$$

$\mathcal{P} : \{\hat{p}, \hat{q}, \hat{f}\}$, *where* $\vec{Y} = (X, Y, Z)$ *and*

$$\begin{aligned} D(\hat{p}) = \{\hat{p}(X, s(n)) &= \hat{p}(X, n) \vee \neg P(X(s(n)), \hat{f}(a, s(n))), \\ \hat{p}(X, \bar{0}) &= \neg P(X(\bar{0}), \hat{f}(a, \bar{0}))\} \end{aligned}$$

$$D(\hat{q}) = \{\hat{q}(X, Y, Z, n, m) = P(\hat{f}(Y(n), m), Z(n)) \wedge \hat{p}(X, n)\}.$$

We extend the calculus RPL_0^Ψ *by two rules involving link-variables. We create link variables* $V_{\hat{p}}(X, r)$ *corresponding to the formulas* $\hat{p}(X, r)$ *for* $r \in \{\bar{0}, n, p(n)\}$ *and add rules for the elimination and the introduction of the variable* $V_{\hat{p}}(X, r)$*:*

$$\frac{V_{\hat{p}}(X, r)}{\vdash \hat{p}(X, r)}\ V_{\hat{p}}E \qquad \frac{\vdash \hat{p}(X, r)}{V_{\hat{p}}(X, r)}\ V_{\hat{p}}I$$

The idea is to use the link variables to define recursive proofs. We start with a proof which is defined for $n > \bar{0}$ *as* $\rho(X, Y, Z, n, m, V_{\hat{p}}(X, n)) =$

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{V_{\hat{p}}(X, n)}{\vdash \hat{p}(X, n)}\ V_{\hat{p}}E}{\vdash \hat{p}(X, p(n)) \vee \neg P(X(n), \hat{f}(a, n))}}{\vdash \hat{p}(X, p(n)), \neg P(X(n), \hat{f}(a, n))}}{P(X(n), \hat{f}(a, n)) \vdash \hat{p}(X, p(n))} \qquad \vdash P(\hat{f}(Y(n), m), Z(n))}{\cfrac{\vdash \hat{p}(X, p(n))}{V_{\hat{p}}(X, p(n))}\ V_{\hat{p}}I}\ R(\Theta(n, m))$$

where $\Theta(n, m) = \{X(n) \leftarrow \hat{f}(Y(n), m), Z(n) \leftarrow \hat{f}(a, n)\}$ *is an s-substitution. The idea is now to append the proof above to itself until we arrive at the sequent* $\vdash \hat{p}(X, \bar{0})$*. We achieve this by the following recursive definition (where* $\hat{\rho}$ *is a recursive proof symbol).*

$$\begin{aligned} &\hat{\rho}(X, Y, Z, n, m, V_{\hat{p}}(X, n)) = \ \underline{\mathrm{if}}\ n = \bar{0}\ \underline{\mathrm{then}}\ \vdash \hat{p}(X, \bar{0}) \\ &\quad \underline{\mathrm{else}}\ \rho(X, Y, Z, n, m, V_{\hat{p}}(X, n)) \circ \hat{\rho}(X, Y, Z, p(n), m, V_{\hat{p}}(X, p(n))). \end{aligned}$$

In carrying out the composition $\rho(X, Y, Z, n, m, V_{\hat{p}}(X, n)) \circ \hat{\rho}(X, Y, Z, p(n), m, V_{\hat{p}}(X, p(n)))$ *we identify the last sequent of* $\rho(X, Y, Z, n, m, V_{\hat{p}}(X, n))$ *with the uppermost leaf* $\vdash V_{\hat{p}}(X, p(n))$ *in* $\hat{\rho}(X, Y, Z, p(n), m, V_{\hat{p}}(X, p(n)))$ *provided* $p(n) > \bar{0}$*, otherwise we end up with the end-sequent* $\vdash \hat{p}(X, \bar{0})$*. Still we do not have a proof with the right axioms, as - for* $n > \bar{0}$ *one axiom in* $\hat{\rho}(X, Y, Z, n, m, V_{\hat{p}}(X, n))$ *is* $\vdash V_{\hat{p}}(X, n)$*. We only have to apply the substitution* $\{V_{\hat{p}}(X, n) \leftarrow \vdash \hat{p}(X, n)$ *to achieve the proof*

$$\hat{\rho}(X, Y, Z, n, m, \hat{p}(X, n))$$

which is a proof of $\vdash \hat{p}(X, \bar{0})$ *from the axioms* $\vdash \hat{p}(X, n)$ *and* $\vdash P(\hat{f}(Y(k), m), Z(k))$ *for* $k \leq n$*. As both* $\vdash \hat{p}(X, n)$ *and the sequents* $\vdash P(\hat{f}(Y(k), m), Z(k))$ *are derivable from* $\vdash \hat{q}(X, Y, Z, n, m)$ *(we also use variable renaming) the following derivation below is a refutation of* $\vdash \hat{q}(X, Y, Z, n, m)$*:* $\rho_0(X, Y, Z, n, m) =$

$$\cfrac{\cfrac{\cfrac{\begin{array}{c}\hat{\rho}(X, Y, Z, n, m, \hat{p}(X, n)) \\ \vdash \hat{p}(X, \bar{0})\end{array}}{\vdash \neg P(X(\bar{0}), \hat{f}(a, \bar{0}))}}{P(X(\bar{0}), f(a, \bar{0})) \vdash} \qquad \vdash P(\hat{f}(Y(0), m), Z(0))}{\vdash}\ R(\Theta(\bar{0}, m))$$

where $\Theta(\bar{0}, m) = \{X(\bar{0}) \leftarrow \hat{f}(Y(0), m), Z(0) \leftarrow \hat{f}(a, \bar{0})\}$ *is an s-substitution. We can also explicitly insert the missing proofs from* $\vdash \hat{q}(X, Y, Z, n, m)$*:*

To obtain a derivation of the leaf $\vdash \hat{p}(X, n)$ *we just define*

$$\hat{\rho}(X, Y, Z, n, m, V_{\hat{p}}(X, n))\{V_{\hat{p}}(X, n) \leftarrow \rho'\}$$

for $\rho' =$

$$\cfrac{\cfrac{\cfrac{\vdash \hat{q}(X, Y, Z, n, m)}{\vdash P(\hat{f}(Y(n), m), Z(n)) \wedge \hat{p}(X, n)}}{\vdash \hat{p}(X, n)}}{V_{\hat{p}}(X, n)}$$

The other proof is just

$$\cfrac{\cfrac{\vdash \hat{q}(X, Y, Z, n, m)}{\vdash P(\hat{f}(Y(0), m), Z(0)) \wedge \hat{p}(X, n)}}{\vdash P(\hat{f}(Y(0), m), Z(0))}$$

Definition 9 (link-variables)**.** *Let* $\hat{p}$ *be a schematic predicate symbol. To* $\hat{p}$ *we assign an infinite set of* link variables $V(\hat{p}) = \{V_i \mid i \in \mathbb{N}\}$*; for different schematic predicate symbols the link variables are disjoint. If* $V \in V(\hat{p})$ *we also say that* V *is of type* $\hat{p}$*. Let* $\hat{p}(\vec{X}, \vec{r})$ *be a schematic atom defined via* $\hat{p}$*. Then, for every* $V \in V(\hat{p})$*, the expression* $V(\vec{X}, \vec{r})$ *is called a* link expression *corresponding to* $\hat{p}$*; we also write* $V_{\hat{p}}(\vec{X}, \vec{r})$ *for this link expression to emphasize that* V *is in* $V(\hat{p})$*. Two link expressions* $V_{\hat{p}}(\vec{X}, \vec{r})$ *and* $U_{\hat{q}}(\vec{Y}, \vec{s})$ *are defined as equal if* $U = V$*,* $\vec{X} = \vec{Y}$ *and* $\vec{r} = \vec{s}$*.*

Link variables V serve the purpose to define locations in a proof where V can be replaced by a proof; these locations can be either the leaves of a proof or the root. In order to place into or to remove variables from proofs we extend the RPL_0^Ψ-calculus by variable elimination rules and variable introduction rules.

Definition 10 ($\mathrm{RPL}_0^\Psi V$)**.** *The calculus* $\mathrm{RPL}_0^\Psi V$ *contains the rules of* RPL_0^Ψ *with two additional rules. Let* V *be a variable of type* $\hat{p}$ *then we define the rules*

$$\frac{\vdash \hat{p}(\vec{X}, \vec{r})}{V(\vec{X}, \vec{r})}\ V_I \quad \frac{V(\vec{X}, \vec{r})}{\vdash \hat{p}(\vec{X}, \vec{r})}\ V_E$$

Elimination rules can only be applied to leaves in an $\mathrm{RPL}_0^\Psi V$ *-derivation (if the leaf is a link expression) and introduction rules to root nodes which are labeled by a sequent of the form* $\vdash \hat{p}(\vec{X}, \vec{r})$*. Or expressed in another way: any* $\mathrm{RPL}_0^\Psi V$*-derivation can be obtained from an* RPL_0^Ψ*- derivation* ρ *by appending variable elimination rules on some leaves of* ρ *and (possibly) a variable introduction rule on the root (provided the sequents on the nodes are of an appropriate form).*

Example 5. *The proof* $\rho(X, Y, Z, n, m, V_{\hat{p}}(X, n))$ *for* $n > \bar{0}$ *in Example 4 is a* $\mathrm{RPL}_0^\Psi V$*-derivation.*

The link variables in an $\mathrm{RPL}_0^\Psi V$-derivation can be replaced by other $\mathrm{RPL}_0^\Psi V$-derivations:

Definition 11 (proof composition)**.** *Let ρ_1 be a $\mathrm{RPL}_0^\Psi V$-derivation with a root node $V_{\hat{p}}(\vec{X},\vec{r})$ and let ρ_2 be a $\mathrm{RPL}_0^\Psi V$-derivation with (possibly several) leaf nodes $V_{\hat{p}}(\vec{X},\vec{r})$ appearing at the set of positions Λ. Then the composition of ρ_1 and ρ_2, denoted as $\rho_1 \circ \rho_2$, is defined as $\rho_2[\rho_1']_\Lambda$ where ρ_1' is the derivation of $\vdash \hat{p}(\vec{X},\vec{r})$, the premise of $V_{\hat{p}}(\vec{X},\vec{r})$ (note that the last rule in ρ_1 is the variable introduction rule for $V_{\hat{p}}(\vec{X},\vec{r})$). ρ_1 and ρ_2 are called* composable *if there exists a proof variable V which is the root node of ρ_1 and a leaf node of ρ_2.*

We did not write $\rho_2\{V_{\hat{p}}(\vec{X},\vec{r}) \leftarrow \rho_1'\}$ for $\rho_1 \circ \rho_2$ because we do not exclude that $V_{\hat{p}}(\vec{X},\vec{r})$ is also the root node of ρ_2.

Example 6. *Let ρ_2 be the proof*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{V_{\hat{p}}(X,n)}{\vdash \hat{p}(X,n)}\ V_{\hat{p}}E}{\vdash \hat{p}(X,p(n)) \vee \neg P(X(n),\hat{f}(a,n))}}{\vdash \hat{p}(X,p(n)), \neg P(X(n),\hat{f}(a,n))}}{P(X(n),\hat{f}(a,n)) \vdash \hat{p}(X,p(n))} \qquad \vdash P(\hat{f}(Y(n),m),Z(n))}{\dfrac{\hat{p}(X,p(n))}{V_{\hat{p}}(X,p(n))}\ V_{\hat{p}}I}\ R(\Theta(n,m))$$

and ρ_1 be

$$\dfrac{\dfrac{\dfrac{\vdash \hat{q}(X,Y,Z,n,m)}{\vdash P(\hat{f}(Y(n),m),Z(n)) \wedge \hat{p}(X,n)}}{\vdash \hat{p}(X,n)}}{V_{\hat{p}}(X,n)}$$

Then $\rho_1 \circ \rho_2$ =

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash \hat{q}(X,Y,Z,n,m)}{\vdash P(\hat{f}(Y(n),m),Z(n)) \wedge \hat{p}(X,n)}}{\vdash \hat{p}(X,n)}}{\vdash \hat{p}(X,p(n)) \vee \neg P(X(n),\hat{f}(a,n))}}{\vdash \hat{p}(X,p(n)), \neg P(X(n),\hat{f}(a,n))}}{P(X(n),\hat{f}(a,n)) \vdash \hat{p}(X,p(n))} \qquad \vdash P(\hat{f}(Y(n),m),Z(n))}{\dfrac{\hat{p}(X,p(n))}{V_{\hat{p}}(X,p(n))}\ V_{\hat{p}}I}\ R(\Theta(n,m))$$

Definition 12 (proof recursion)**.** *Let $\rho(\bar{X},\vec{n},V_{\hat{p}}(\vec{Y},\vec{m},k))$ be a proof with one or several leaves $V_{\hat{p}}(\vec{Y},\vec{m},k)$ and with the root $V_{\hat{p}}(\vec{Y},\vec{m},p(k))$, where $\vec{Y}$ is a subvector of $\vec{X}$ and $(\vec{m},k)$ of $\vec{n}$ (if this is the case we say that ρ admits proof recursion). We abbreviate $V_{\hat{p}}(\vec{Y},\vec{m},k)$ by $V(k)$ and define*

$$\underline{\text{if}}\ k = \bar{0}\ \underline{\text{then}}\ \hat{\rho}((\bar{X},\vec{n},V(k)) = V(\bar{0})$$
$$\underline{\text{else}}\ \hat{\rho}((\bar{X},\vec{n},V(k)) = \rho(\bar{X},\vec{n},V(k)) \circ \hat{\rho}((\bar{X},\vec{n}\{k \leftarrow p(k)\},V(p(k)))).$$

Note that from $V(\bar{0})$ we can finally derive $\hat{p}(\vec{Y},\vec{m},\bar{0})$. We say that $\hat{\rho}$ is the inductive closure *of ρ.*

Example 7. *Take* $\rho(X, Y, Z, n, m, V_{\hat{p}}(X, n))$ *and* $\hat{\rho}(X, Y, Z, n, m, V_{\hat{p}}(X, n))$ *from Example 4. Then* $\hat{\rho}$ *is the inductive closure of* ρ.

Note that $\hat{\rho}$ is obtained from ρ by a kind of primitive recursion on proofs.

Definition 13 (proof schema)**.** *We define proof schema inductively:*

- *Any* $\mathrm{RPL}_0^{\Psi}V$*-derivation is a proof schema.*
- *If* ρ_1 *and* ρ_2 *are proof schemata and* ρ_1, ρ_2 *are composable then* $\rho_1 \circ \rho_2$ *is a proof schema.*
- *If* ρ *is a proof schema which admits proof recursion then the inductive closure of* ρ *is a proof schema.*
- *Let* $\rho_1(\vec{X_1}, \vec{n}), \ldots, \rho_\alpha(\vec{X_\alpha}, \vec{n})$ *(for* $\alpha > 0$*) be proof schemata over the parameter tuple* $\vec{n}$ *and let* $\{C_1, \ldots, C_\alpha\}$ *be conditions on the parameters in* $\vec{n}$ *which define a partition then*

$$\begin{array}{l}\underline{\text{if}}\ C_1\ \underline{\text{then}}\ \rho_1(\vec{X_1}, \vec{n})\ \underline{\text{else}}\\ \ \underline{\text{if}}\ C_2\ \underline{\text{then}}\ \rho_2(\vec{X_2}, \vec{n})\ \underline{\text{else}}\\ \ldots\\ \quad \underline{\text{if}}\ C_{\alpha-1}\ \underline{\text{then}}\ \rho_{\alpha-1}(\vec{X_{\alpha-1}}, \vec{n})\ \underline{\text{else}}\ \rho_\alpha(\vec{X_\alpha}, \vec{n})\end{array}$$

  is a proof schema.

Example 8. *Consider the proofs* ρ, $\hat{\rho}$ *and* ρ_0 *in Example 4. All of them are proof schemata; the proof schema* ρ_0 *is also a* refutation schema *of* $\hat{q}(X, Y, Z, n, m)$*, a concept which will be formally defined below.*

Our schematic proofs are proofs from sequents of the form $\vdash F$, which we call schematic F-proofs. A schematic F-proof of $\vdash$ is called a refutation schema of F.

Definition 14 (schematic F-proofs)**.** *Let* $F\colon \hat{q}(\vec{X}, \vec{n})$ *be the main schematic atom in a schematic definition* Ψ*. We define schematic* F*-proofs below. The* $\mathrm{RPL}_0^{\Psi}V$*-proof*

$$\vdash \hat{q}(\vec{X}, \vec{n})$$

is a schematic F*-proof.*

- *If* ρ_1 *and* ρ_2 *are schematic* F*-proofs of* S_1 *and* S_2 *from* $\vdash \hat{q}(\vec{X}, \vec{n})$ *and* $\rho =$

$$\dfrac{\begin{array}{cc}(\rho_1) & (\rho_2)\\ S_1 & S_2\end{array}}{S}\ \xi$$

  for a binary rule ξ *then* ρ *is a schematic* F*-proof of* S.

- *If* ρ' *is a schematic* F*-proof of* S' *and* $\rho =$

$$\dfrac{\begin{array}{c}(\rho')\\ S'\end{array}}{S}\ \xi$$

  for a unary rule ξ *then* ρ *is a schematic* F*-proof of* S.

- *Let* ρ_1 *be a schematic* F*-proof of* S *where* S *is of the form* $V_{\hat{p}}(\vec{Z},\vec{k})$*. Let* ρ_2 *be a proof schema with one or several leaves* $\lambda : V_{\hat{p}}(\vec{Z},\vec{k})$ *such that for all other leaves* λ *of* ρ_2 *there are schematic* F*-proofs of* $\mathrm{seq}(\lambda)$*. Then* $\rho_1 \circ \rho_2$ *is a schematic* F*-proof of the end-sequent of* S_2*.*

- *Let* $\rho(\bar{X},\vec{n},V_{\hat{p}}(\vec{Y},\vec{m},k))$ *be a proof schema with a leaf* $V_{\hat{p}}(\vec{Y},\vec{m},k)$ *and with the root* $V_{\hat{p}}(\vec{Y},\vec{m},p(k))$*, where* $\vec{Y}$ *is a subvector of* $\vec{X}$ *and* $(\vec{m},k)$ *of* $\vec{n}$ *and assume that* ρ *admits proof recursion, i.e. it is a proof schema of* $V_{\hat{p}}(\vec{Y},\vec{m},p(k))$ *from* $V_{\hat{p}}(\vec{Y},\vec{m},p(k))$*. Assume further that* $\hat{\rho}$ *is defined as (*$V(k)$ *stands for* $V_{\hat{p}}(\vec{Y},\vec{m},k)$*)*

$$\underline{\text{if}}\ k=\bar{0}\ \underline{\text{then}}\ \hat{\rho}((\bar{X},\vec{n},V(k))=V(\bar{0})$$
$$\underline{\text{else}}\ \hat{\rho}((\bar{X},\vec{n},V(k))=\rho(\bar{X},\vec{n},V(k))\circ\hat{\rho}((\bar{X},\vec{n}\{k\leftarrow p(k)\},V(p(k)))).$$

If, for $l \le k$*, the proof schema* $\rho(\bar{X},\vec{n}\{k\leftarrow l\},V_{\hat{p}}(\vec{Y},\vec{m},l))$ *is a schematic* F*-proof (of its end-sequent) then* $\hat{\rho}((\bar{X},\vec{n},V(k))$ *is a schematic* F*-proof of* $V(\bar{0})$*.*

- *Let* $\rho_1(\vec{X_1},\vec{n}),\rho_2(\vec{X_2},\vec{n})$ *be schematic* F*-proofs over the parameter tuple* $\vec{n}$ *and let* C *be a condition on the parameters in* $\vec{n}$*; let* $\rho(\vec{X_1},\vec{X_2},\vec{n}) =$

$$\underline{\text{if}}\ C\ \underline{\text{then}}\ \rho_1(\vec{X_1},\vec{n})\ \underline{\text{else}}\ \rho_2(\vec{X_2},\vec{n})$$

Then $\rho(\vec{X_1},\vec{X_2},\vec{n})$ *is a schematic* F*-proof of* S*.*

Example 9. *The proof schema* ρ_0 *in Example 4 is a refutation schema of* $\hat{q}(X,Y,Z,n,m)$*.*

When the parameters in a refutation schema are instantiated with numerals, we obtain a $\mathrm{RPL}_0^{\Psi}V$ refutation.

Theorem 1. *Let* ρ *be a refutation schema of a schematic atom* $\hat{q}(\vec{X},n_1,\dots,n_\alpha)$*. Then, for all numerals* $\nu_1,\dots,\nu_\alpha$*, the evaluation of* $\rho\{n_1\leftarrow\nu_1,\dots,n_\alpha\leftarrow\nu_\alpha\}$ *is a* $\mathrm{RPL}_0^{\Psi}V$ *refutation of* $\hat{q}(\vec{X},\nu_1,\dots,\nu_\alpha)$*.*

Proof. Instantiate all parameters in ρ, and replace proof recursions by the corresponding derivations, according to the parameter instantiations. Proof recursions are defined in a primitive recursive way, and eventually reach a base case. What is left, is a derivation from instances of $\vdash \hat{q}(\vec{X},n_1,\dots,n_\alpha)$ of the empty sequent in $\mathrm{RPL}_0^{\Psi}V$. $\mathrm{RPL}_0^{\Psi}V$ is trivially sound. □

Using the above formalism for refutation schemata, it is possible to extract a schematic structure representing the Herbrand sequent of the refutation.

Example 10. *Consider the refutation schema* ρ_0*, as defined in Example 4.* ρ_0 *defines the s-substitution*

$$\Theta(\bar{0},m)=\{X(\bar{0})\leftarrow\hat{f}(Y(0),m),Z(0)\leftarrow\hat{f}(a,\bar{0})\}.$$

Moreover, as ρ_0 *contains the recursive proof symbol* $\hat{\rho}$ *as axiom, we have to take the substitutions coming from* $\hat{\rho}$ *into account as well! In fact,* $\Theta(\bar{0},m)$ *will be applied to all the other s-substitutions coming from the rule applications "above", i.e. to all the substitutions in derivations corresponding to the proof variables in the leaves.* $\hat{\rho}$ *is recursively defined over the derivation* ρ*, which defines the s-substitution*

$$\Theta(n,m)=\{X(n)\leftarrow\hat{f}(Y(0),m),Z(0)\leftarrow\hat{f}(a,n)\}.$$

By construction, as $\hat{\rho}$ is recursive, $\Theta(p(n), m)$ is applied to $\Theta(n, m)$; to the result we apply $\Theta(p(p(n)), m)$ and so on. Intuitively, $\Theta^*(n, m)$ is is the sequence

$$\Theta(n, m)\Theta(p(n), m)\Theta(p(p(n)), m)\cdots\Theta(\bar{0}, m).$$

Hence, we obtain the Herbrand schema

$$\Theta^*(n, m) = \{\Theta(0, m) : n = 0, \quad \Theta(n, m)\Theta^*(p(n), m) : n > 0\}.$$

Notice that the variables in the s-substitutions $\Theta(n, m), \Theta(n-1, m), \ldots$ are different, and therefore the application of $\Theta(\bar{0}, m)$ to $\Theta(\bar{1}, m)$, and so on, results in the union of the substitutions.

Definition 15 (Herbrand schema)**.** *The Herbrand schema of a refutation schema is defined inductively as:*

- *For $\vdash F$ the Herbrand schema is $\{\emptyset\}$, the set containing the identical substitution.*
- *Let ρ be a schematic F-proof of the form*

$$\cfrac{\begin{matrix}(\rho_1) & (\rho_2)\\ S_1 & S_2\end{matrix}}{S}\ \xi$$

  and assume the Herbrand schema of ρ_1 and ρ_2 are Θ_1 and Θ_2. If ξ is a binary rule different to the resolution rule, then the Herbrand schema of ρ is the global s-unifier of $\Theta_1 \cup \Theta_2$, which can be computed after regularization of the proof. If ξ is a resolution rule of the form $R(\Theta)$, then prior regularization is mandatory and the Herbrand schema is $(\Theta_1 \cup \Theta_2) \circ \Theta$.

- *Let ρ be a schematic F-proof of the form*

$$\cfrac{\begin{matrix}(\rho')\\ S'\end{matrix}}{S}\ \xi$$

  and assume that the Herbrand schema of ρ' is Θ, then the Herbrand schema of ρ is Θ.

- *Let ρ be of the form $\rho_1 \circ \rho_2$, then the Herbrand schema of ρ is $\Theta_1 \circ \Theta_2$, where Θ_1 is the Herbrand schema of ρ_1, and Θ_2 is the Herbrand schema of ρ_2,*
- *Let $\rho(\bar{X}, \vec{n}, V_{\hat{p}}(\vec{Y}, \vec{m}, k))$ be a proof schema of $V_{\hat{p}}(\vec{Y}, \vec{m}, p(k))$ from $V_{\hat{p}}(\vec{Y}, \vec{m}, k)$, let $\Theta(\vec{n})$ be the global s-unifier of the derivation, and let $\hat{\rho}$ be defined as ($V(k)$ stands for $V_{\hat{p}}(\vec{Y}, \vec{m}, k)$)*

$$\begin{array}{l}\underline{\text{if}}\ k = \bar{0}\ \underline{\text{then}}\ \hat{\rho}((\bar{X}, \vec{n}, V(k)) = V(\bar{0})\\ \underline{\text{else}}\ \hat{\rho}((\bar{X}, \vec{n}, V(k)) = \rho(\bar{X}, \vec{n}, V(k)) \circ \hat{\rho}((\bar{X}, \vec{n}\{k \leftarrow p(k)\}, V(p(k)))).\end{array}$$

  To ensure that $\hat{\rho}$ is regular we must ensure that all variable expressions in ρ contain the parameter n. If this is not the case the variables need to be renamed. Then the Herbrand schema of ρ is defined as

$$\begin{array}{l}\underline{\text{if}}\ k = \bar{0}\ \underline{\text{then}}\ \Theta^*(\vec{n}) = \Theta(\vec{n})\\ \underline{\text{else}}\ \Theta^*(\vec{n}) = \Theta(\vec{n}) \circ \Theta^*(\vec{n}\{k \leftarrow p(k)\}).\end{array}$$

- *Let* $\rho(\vec{X}_1, \vec{X}_2, \vec{n}) =$
$$\underline{\text{if}}\ C\ \underline{\text{then}}\ \rho_1(\vec{X}_1, \vec{n})\ \underline{\text{else}}\ \rho_2(\vec{X}_2, \vec{n}).$$
be a schematic F-proof, and assume that the Herband schema of ρ_1, ρ_2 are Θ_1, Θ_2. Then the Herbrand schema of ρ is
$$\underline{\text{if}}\ C\ \underline{\text{then}}\ \Theta_1\ \underline{\text{else}}\ \Theta_2.$$

Example 11. *Consider the Herbrand schema from Example 10 and the fixed parameters $n = 1$ and $m = 0$. Then,*

$$\begin{aligned}
\Theta^*(1,0) &= \Theta(1,0)\Theta^*(p(1),0) \\
&= \{\{X(1) \leftarrow \hat{f}(Y(1),0), Z(1) \leftarrow \hat{f}(a,1)\}\}\{\{X(0) \leftarrow \hat{f}(Y(0),0), Z(0) \leftarrow \hat{f}(a,0)\}\} \\
&= \{\{X(1) \leftarrow Y(1),\ Z(1) \leftarrow f(a),\ X(0) \leftarrow Y(0),\ Z(0) \leftarrow a\}\}.
\end{aligned}$$

Applying $\Theta^*(1,0)$ to the initial sequents results in an unsatisfiable set of sequents.

Theorem 2. *Let Θ be a Herbrand schema of a refutation schema ρ of $\hat{q}(\vec{X}, n_1, \ldots, n_\alpha)$. Then, for all numerals $\nu_1, \ldots, \nu_\alpha$, the set of sequents*

$$\{\vdash \hat{q}(\vec{X}, n_1, \ldots, n_\alpha)\Theta\}\{n_1 \leftarrow \nu_1, \ldots, n_\alpha \leftarrow \nu_\alpha\}$$

is unsatisfiable.

Proof. For all numerals, first apply the s-substitutions defined in Θ to ρ. The initial sequents are then instances of $\vdash \hat{q}(\vec{X}, n_1, \ldots, n_\alpha)$, and the resolution rules turn into applications of cut rules. The thus obtained derivation is in RPL_0^Ψ, where the resolution rules are cut rules. As this calculus is sound the set of sequents

$$\{\vdash \hat{q}(\vec{X}, n_1, \ldots, n_\alpha)\Theta\}\{n_1 \leftarrow \nu_1, \ldots, n_\alpha \leftarrow \nu_\alpha\}$$

occurring at the leaves is unsatisfiable. □

The formalism for schematic proofs presented in this paper is more powerful than that defined in [12] and in [7] as these former approaches missed the inductive closure of proofs as a syntactic proof object. In fact inductive closures could only be computed for given parameter assignments, a representation on the syntax was missing. As a consequence there was no way to define Herbrand schemata in a general way. Moreover, the proof recursion defines new schematic proofs which can be called from others and thus considerably extends the expressivity of the former approaches.

5 Conclusion

We introduced the calculus $\mathrm{RPL}_0^\Psi V$ for the construction of a schematic refutation of a quantifier-free formula schema. This formula schema originates from the proof analysis method CERES, where it defines the derivations of the cut formulas in the original proof. We have shown that from the refutation schema of this formula schema, a Herbrand schema can be constructed, which can be used to compute the Herbrand schema of the original proof schema containing cuts. The formalism for proof schemata as presented in this work is new, extends the expressivity, and simplifies existing notions of proof as schema.

References

[1] Peter B. Andrews. Resolution in type theory. *J. Symb. Log.*, 36(3):414–432, 1971.

[2] Vincent Aravantinos, Ricardo Caferra, and Nicolas Peltier. Decidability and undecidability results for propositional schemata. *J. Artif. Intell. Res.*, 40:599–656, 2011.

[3] Vincent Aravantinos, Mnacho Echenim, and Nicolas Peltier. A resolution calculus for first-order schemata. *Fundam. Informaticae*, 125(2):101–133, 2013.

[4] 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.

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

[6] Matthias Baaz and Alexander Leitsch. *Methods of Cut-elimination*, volume 34. Springer Science & Business Media, 2011.

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

[8] Jacques Herbrand. *Recherches sur la théorie de la démonstration.* PhD thesis, Université de Paris, 1930.

[9] Stefan Hetzl, Alexander Leitsch, Daniel Weller, and Bruno Woltzenlogel Paleo. Herbrand sequent extraction. In *AISC/MKM/Calculemus*, volume 5144 of *Lecture Notes in Computer Science*, pages 462–477. Springer, 2008.

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

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

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