Journal of Automated Reasoning (2021) 65:599–645
https://doi.org/10.1007/s10817-020-09583-8

Schematic Refutations of Formula Schemata

David M. Cerna[1,2] · Alexander Leitsch[3] · Anela Lolic[3]

Received: 19 December 2019 / Accepted: 29 October 2020 / Published online: 19 November 2020

Abstract
Proof schemata are infinite sequences of proofs which are defined inductively. In this paper we present a general framework for schemata of terms, formulas and unifiers and define a resolution calculus for schemata of quantifier-free formulas. The new calculus generalizes and improves former approaches to schematic deduction. As an application of the method we present a schematic refutation formalizing a proof of a weak form of the pigeon hole principle.

Keywords Schematic proofs · Resolution · Induction · Schematic formulas

1 Introduction

Recursive definitions of functions play a central role in computer science, particularly in functional programming. While recursive definitions of proofs are less common they are of increasing importance in automated proof analysis. Proof schemata, i.e. recursively defined infinite sequences of proofs, serve as an alternative formulation of induction. Prior to the formalization of the concept, an analysis of Fürstenberg's proof of the infinitude of primes [5] suggested the need for a formalism quite close to the type of proof schemata we will discuss in this paper. The underlying method for this analysis was CERES [6] (cut-elimination by resolution) which, unlike reductive cut-elimination, can be applied to recursively defined proofs by extracting a schematic unsatisfiable formula and constructing a recursively defined refutation. Moreover, Herbrand's theorem can be extended to an expressive fragment of proof schemata, that is those formalizing k-induction [11,14]. Unfortunately, the construction of recursively defined refutations is a highly complex task. In previous work [14] a superposition

✉ Anela Lolic
anela@logic.at

David M. Cerna
dcerna@cas.cs.cz; david.cerna@risc.jku.at

Alexander Leitsch
leitsch@logic.at

[1] The Czech Academy of Science, Institute of Computer Science (CAS ICS), Prague, Czechia

[2] Research Institute for Symbolic Computation, Johannes Kepler University, Linz, Austria

[3] Institute of Logic and Computation, TU Wien, Vienna, Austria

calculus for certain types of formulas was used for the construction of refutation schemata, but only works for a weak fragment of arithmetic and is hard to use interactively.

The key to proof analysis using CERES in a first-order setting is not the particularities of the method itself, but the fact that it provides a bridge between automated deduction and proof theory. In the schematic setting, where the proofs are recursively defined, a bridge over the chasm has been provided [11,14], but there has not been much development on the other side to reap the benefits of. The few existing results about automated deduction for recursively defined formulas barely provide the necessary expressive power to analyse significant mathematical argumentation. Applying the earlier constructions to a weak mathematical statement such as the *eventually constant schema* required much more work than the value of the provided insights [10]. The resolution calculus we introduce in this work generalizes resolution and the first-order language in such a way that it provides an excellent environment for carrying out investigations into decidable fragments of schematic propositional formulas beyond those that are known. Furthermore, concerning the general unsatisfiability problem for schematic formulas, our formalism provides a perfect setting for interactive proof construction.

Proof schema is not the first alternative formalization of induction with respect to Peano arithmetic [17]. However, all other existing examples [8,9,15] that provide calculi for induction together with a cut-elimination procedure do not allow the extraction of Herbrand sequents[1] [12,17] and thus Herbrand's theorem cannot be realized. In contrast, in [14] finite representations of infinite sequences of Herbrand sequents are constructed, so-called *Herbrand systems*. Of course, such objects do not describe finite sets of ground instances, though instantiating the free parameters (i.e. variables that can be instantiated with numerals) of Herbrand systems does result in sequents derivable from a finite set of ground instances.

The formalism developed in this paper extends and improves the formal framework for refuting formula schemata in [11,14] in several ways: 1. The new calculus can deal with arbitrary quantifier-free formula schemata (not only with clause schemata), 2. we extend the schematic formalism to multiple parameters (in [11] and in [14] only schemata defined via one parameter were admitted); 3. we strongly extend the recursive proof specifications by allowing mutual recursion (formalizable by so-called called point transition systems). Note that in [11] a complicated schematic clause definition was used, while the schematic refutations in [14] were based on negation normal forms and on a complicated translation to the n-clause calculus. Moreover, the new method presented in this paper provides a simple, powerful and elegant formalism for interactive use. The expressivity of the method is illustrated by an application to a (weak) version of the pigeon hole principle.

2 A Motivational Example

In [10], proof analysis of a mathematically simple statement, the *Eventually Constant Schema*, was performed using an early formalism developed for schematic proof analysis [11]. The Eventually Constant Schema states that any monotonically decreasing function with a finite range is eventually constant. The property of being eventually constant may be formally written as follows:

$$\exists x \forall y (x \leq y \rightarrow f(x) = f(y)), \tag{1}$$

[1] Herbrand sequents allow the representation of the propositional content of first-order proofs.

where f is an uninterpreted function symbol with the following property

$$\forall x \left(\bigvee_{i=0}^{n} f(x) = i \right)$$

for some $n \in \mathbb{N}$. The method defined in [11] requires a strong quantifier-free end sequent, thus implying the proof must be skolemized. The skolemized formulation of the eventually constant property is $\exists x(x \leq g(x) \rightarrow f(x) = f(g(x)))$ where g is the introduced Skolem function. The proof presented in [10] used a sequence of Σ_2-cuts

$$\exists x \forall y(((x \leq y) \Rightarrow n + 1 = f(y)) \vee f(y) < n + 1).$$

Also, the Skolem function was left uninterpreted for the proof analysis. The resulting cut-structure, when extracted as an unsatisfiable clause set, has a fairly simple refutation. Thus, with the aid of automated theorem provers, a schema of refutations was constructed.

The use of an uninterpreted Skolem function greatly simplified the construction presented in [10]. In this paper we will interpret the function g as the successor function. Note that using the axioms presented in [10] the following statement

$$\forall x \left(\bigvee_{i=0}^{n} f(x) = i \right) \vdash \exists x(f(x) = f(suc(x)))$$

is not provable. Note that we drop the implication of Equation 1 and the antecedent of the implication given that $x \leq suc(x)$ is a trivial property of the successor function. However, using an alternative set of axioms and a weaker cut we can prove this statement. The additional axioms are as follows:

$$\begin{aligned}
f(x) = i &\vdash f(x) < s(k)\ , \text{ for } 0 \leq i \leq k < n \\
f(suc(x)) = i &\vdash f(x) < s(k)\ , \text{ for } 0 \leq i \leq k < n \\
f(x) = k, f(suc(x)) = k &\vdash f(x) = f(suc(x))\ , \text{ for } 0 \leq k \leq n \\
f(0) < 0 &\vdash \\
f(suc(x)) < s(k) &\vdash f(suc(x)) = k, f(x) < k\ , \text{ for } 0 \leq k \leq n \\
f(x) < s(k) &\vdash f(x) = k, f(x) < k, \text{ for } 0 \leq k < n
\end{aligned}$$

For the most part these axioms are harmless, however the axiom $f(suc(x)) < s(k) \vdash f(suc(x)) = k, f(x) < k$ implies that f has some monotonicity properties similar to the eventually constant schema.

Note that the mentioned axiom $f(x) < s(k) \vdash f(x) = k, f(x) < k$ is equivalent to

$$f(k) \geq k \vdash f(suc(x)) \geq k$$

in the standard model. Thus this axiom describes an *increase* of values, *not a decrease*! For example consider the following interpretation of f for $n = 2$:

$$f(0) = 0,\ f(1) = 1,\ f(2) = 2,\ f(z) = 2 \text{ for all } z > 1.$$

Here we have $f(1) < 2$, but $f(2) = 2$ and $f(z) = 2$ for all $z > 1$.

Being that our proof enforces this property using the following Δ_2-cut formula we are guaranteed to reach a value in the domain above which f is constant. The cut formula is:

$$\exists x(f(x) = k \wedge k = f(suc(x))) \vee \forall x(f(x) < k)\ , \text{ for } 0 \leq k \leq n.$$

```
============================== PROOF =================================

% Proof 1 at 0.017 (+ 0.000) seconds.
% Given clauses 73.
% number of calls to fixpoint : 3
 S_init  :
 (51:  [ EQ(v0,f(h(v1))) | LE(f(v1),v0) if  n = s(v0) ].
50:  [ EQ(v0,f(v1)) | LE(f(v1),v0) if  n = s(v0) ].
33:  [ PHI(v0,v1) if  n = s(v0) ].
)
 S_loop  :
 (82:  [ EQ(v0,f(h(v1))) | LE(f(v1),v0) if  n = s(s(v0)) ].
80:  [ EQ(v0,f(v1)) | LE(f(v1),v0) if  n = s(s(v0)) ].
53:  [ PHI(v0,v1) if  n = s(s(v0)) ].
)
 The empty clauses  :
 (45:  [  n = 0 ].
81:  [  n = s(0) ].
112:  [  n = s(s(0)) ].
) max_rank 2

============================== end of proof ==========================
```

Fig. 1 Output of Peltier et al.'s Prover9 extension [13]

One additional point which the reader might notice is that we use what seems to be the less than relation and equality relation of the natural numbers, but do not concern ourselves with substitutivity of equality nor transitivity of $<$. While including these properties will change the formal proof we present below, the argument will still require a free numeric parameter denoting the size of the range of f and the number of positions we require to map to the same value.

We will refer to this version of the eventually constant schema as the *successor eventually constant schema*. While this results in a new formulation of the eventually constant schema under an interpretation of the Skolem function as the successor function, we have not taken complete advantage of this new interpretation taking into account that this re-formulation is actually of lower complexity than the eventually constant schema. For example in Fig. 1 we provide the output of Peltier's superposition induction prover [3] when ran on the clausified form of the cut structure of the successor eventually constant schema. The existence of this derivation implies that the proof analysis method of [14] may be applied to the successor eventually constant schema. Unfortunately, the prover does not find the invariant discovered in [10], but this may have more to do with the choice of axioms rather than the statement being beyond the capabilities of the prover.

We can strengthen the successor eventually constant schema beyond the capabilities of [13] easily by adding a second parameter as follows:

$$\forall x \left(\bigvee_{i=0}^{n} f(x) = i \right) \vdash \exists x \left(\bigwedge_{i=0}^{m} f(x) = f(suc^{i}(x)) \right).$$

We refer to this problem as the m-successor eventually constant schema. Applying this transformation to the eventually constant schema of [10] is not so trivial being that the axioms used to construct the proof do not easy generalize. However, for the successor eventually constant schema the generalization is trivial and is provided below:

$f(suc^r(x)) = i \vdash f(x) < s(k)$, for $0 \leq i \leq k \leq n$ and $0 \leq r \leq m$,
$f(x) = k, f(suc^r(x)) = k \vdash f(x) = f(suc^r(x))$, for $0 \leq k \leq n$, and $0 < r \leq m$,
$f(0) < 0 \vdash$,
$f(suc^r(x)) < s(k) \vdash f(suc^r(x)) = k, f(x) < k$, for $0 \leq k \leq n$ and $0 \leq r \leq m$,
where $suc^0(x) = x$.

Similar to the previous case, the last axiom may be interpreted as

$$f(x) \geq k \vdash f(suc^r(x)) \geq k \text{ for } 0 \leq k \leq n,\ 0 \leq r \leq m$$

over the standard model, where $suc^r(x) = x + r$. Again, it describes an increase of values. Furthermore, the cut formula can be trivially extended as follows:

$$\exists x \left(\bigwedge_{i=0}^{m} f(suc^i(x)) = k \right) \vee \forall x (f(x) < k) \ , \text{ for } 0 \leq k \leq n.$$

Given that the m-successor eventually constant schema contains two parameters it is beyond the capabilities of [13]. Interesting enough, the prover can find invariants for each value of m in terms of n, though, these invariants get impressively large quite quickly. The cut structure of the m-successor eventually constant schema may be extracted as an inductive definition of an unsatisfiable negation normal form formula. We provide this definition below:

$$\begin{aligned}
O(n, m) &= D(n, m) \wedge\ P(n, m),\\
C(y, n, 0) &= f(S(0, y)) \nsim n,\\
C(y, n, s(m)) &= f(S(s(m), y)) \nsim n\ \vee\ C(y, n, m),\\
T(n, 0) &= \forall x (f(S(0, x)) \nless s(n)\ \vee\ f(S(0, x)) \sim n\ \vee\ f(x) < n),\\
T(n, s(m)) &= \forall x (f(S(s(m), x)) \nless s(n)\ \vee\ f(S(s(m), x)) \sim n \vee f(x) < n)\\
&\quad \wedge T(n, m),\\
P(0, m) &= \forall x (C(x, 0, m))\ \wedge\ f(a) \nless 0,\\
P(s(n), m) &= \forall x C(x, s(n), m)\ \wedge T(n, m)\ \wedge\ P(n, m),\\
D(n, 0) &= \forall x (f(S(0, x)) \sim n\ \vee\ f(x) < n),\\
D(n, s(m)) &= \forall x (f(S(s(m), x)) \sim n\ \vee\ f(x) < n) \wedge\ D(n, m),\\
S(0, y) &= y,\\
S(s(n), y) &= suc(S(n, y)).
\end{aligned}$$

where a is some arbitrary constant. We will show how our new formalism can provide a finite representation of the refutations of the inductive definition even though our refutation requires the use of mutual recursion as well as multiple parameters (six in total).

3 Schematic Language

Large parts of mathematics can be formalized in a natural way in second-order logic [16]. However, most methods for proof analysis and transformation are particularly suited for first-order logic and thus, second-order formalizations have to be projected to first-order ones. The most appropriate way to deal with this projection is the introduction of a many-sorted language. As in [11,14] we choose to work in a two-sorted version of classical first-order logic with one sort for a standard first-order term language and one for numerals.

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 follows $N \Rightarrow s(N) \mid 0$, s.t. $s(N) \neq 0$ and $s(N) = s(N') \rightarrow N = N'$. Natural numbers ($\mathbb{N}$) will be denoted by lower-case Greek letters (α, β, γ, etc); The numeral $s^\alpha 0, \alpha \in \mathbb{N}$, will be written as $\bar{\alpha}$. The set of numerals is denoted by *Num*. When describing sequences of objects such as $t_1, \cdots, t_\alpha$, if it is possible to avoid confusion, we will abbreviate the sequence by $\overrightarrow{t}_\alpha$.

Furthermore, 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 follows:

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

In addition to the signature Σ_ω, 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 $\hat{f}(\overrightarrow{t}_\alpha) \in T^\omega$

The second sort, the ι-sort (individuals), also has two associated signatures, the set of free function symbols, Σ_ι, and the set of *defined function symbols*, $\hat{\Sigma}_\iota$. Similarly, defined symbols will be denoted by $\hat{\cdot}$ and have a fixed finite arity. Variables of the ι-sort are what we refer to as *global* variables, that is 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. A similar construction will be used for other types of objects defined in this section.

Thus, the set of free ι-terms, denoted by $\mathcal{T}_0^\iota$ is inductively constructed from Σ_ι and V^G as follows:

- If $\bar{\alpha}_1, \cdots, \bar{\alpha}_\beta \in Num$ and $X \in V^G$, then $X(\overrightarrow{\bar{\alpha}_\beta}) \in \mathcal{T}_0^\iota$
- If $t_1, \cdots, t_\alpha \in \mathcal{T}_0^\iota$ and $f \in \Sigma_\iota$ s.t. f has arity $\alpha \geq 0$, then $f(\overrightarrow{t}_\alpha) \in \mathcal{T}_0^\iota$

The set of ι-terms, denoted by $\mathcal{T}^\iota$ is inductively constructed from Σ_ι, $\hat{\Sigma}_\iota$, and V^G as follows:

- If $t \in \mathcal{T}_0^\iota$, then $t \in \mathcal{T}^\iota$
- If $t_1, \cdots, t_\alpha \in \mathcal{T}^\omega$ and $X \in V^G$, then $X(\overrightarrow{t_\alpha}) \in \mathcal{T}^\iota$
- If $t_1, \cdots, t_\alpha \in \mathcal{T}^\iota$, $\hat{f} \in \hat{\Sigma}_\iota$, $\overrightarrow{X_\beta} \in V^G$, and $\overrightarrow{n_{\alpha+1}} \in \mathcal{N}$ s.t. $\hat{f}$ has arity $\alpha + \beta + 1$ for $\alpha, \beta \geq 0$, then $\hat{f}(\overrightarrow{X_\beta}, \overrightarrow{n_{\alpha+1}}) \in \mathcal{T}^\iota$

Remark 1 In this work we will define schematic refutations and schematic unifiers. In previous work (see [14]) in principle only an implicit representation of unification schemata could be obtained, an explicit representation was impossible due to the restrictions of the formalism. If we however allow for the use of indexed variables, we obtain a stronger formalism in the sense that schematic variables, and thus unifiers, can be defined. The use of global variables will be vital for the definition of schematic substitutions and unifiers later in the paper.

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 formula terms, denoted by $\mathcal{T}_V^o$ is constructed inductively as follows:

- if $t \in V^F$, then $t \in \mathcal{T}_V^o$
- if $t_1, \ldots, t_\alpha \in T^\iota$ and $P \in \mathcal{P}$ s.t. P has arity $\alpha \geq 0$, then $P(\overrightarrow{t_\alpha}) \in \mathcal{T}_V^o$.
- 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 $t_1 \star t_2 \in \mathcal{T}_V^o$

We refer to *Boolean expressions* as the subset of $\mathcal{T}_V^o$ constructed without symbols of V^F. For $t \in \mathcal{T}_V^o$, by $V^F(t) \subset V^F$ we denote the set of formula variables occurring in t. The set of Boolean expressions will be denoted by $\mathcal{T}_0^o$ and is constructed as follows:

- if $t_1, \ldots, t_\alpha \in T^\iota$ and $P \in \mathcal{P}$ s.t. P has arity $\alpha \geq 0$, then $P(\overrightarrow{t_\alpha}) \in \mathcal{T}_0^o$.
- if $t \in \mathcal{T}_0^o$, then $\neg t \in \mathcal{T}_0^o$
- if $t_1, t_2 \in \mathcal{T}_0^o$ and $\star \in \{\vee, \wedge\}$, then $t_1 \star t_2 \in \mathcal{T}_0^o$

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

- 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 $t_1 \star t_2 \in \mathcal{T}^o$

Furthermore, for $x \in \{\omega, \iota, o\}$, $\hat{\Sigma}_x$ has an associated irreflexive, transitive, and Noetherian order $<_x$.

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

Definition 1 *(Defining equations)* Let $x \in \{\omega, \iota, o\}$, $\alpha, \beta \geq 0$, and $\bullet$ is a member of $\mathcal{N}$, V^ι, or V^F depending on x. For every $\hat{f} \in \Sigma_x$, we define a set $D(\hat{f})$ consisting of two equations:

$$\hat{f}(\overrightarrow{X}_\alpha, \overrightarrow{n}_\beta, \bar{0}) = t_B^{\hat{f}}, \quad \hat{f}(\overrightarrow{X}_\alpha \overrightarrow{n}_\beta, s(m)) = t_S^{\hat{f}}\{\bullet \leftarrow \hat{f}(\overrightarrow{X}_\alpha \overrightarrow{n}_\beta, m)\}, \text{ where}$$

(1) If $\hat{f}$ is minimal:

(a) if $x \in \{\omega, \iota\}$, then $t_B^{\hat{f}}, t_S^{\hat{f}} \in T_0^x$

(b) if $x = o$, then $t_B^{\hat{f}} \in T_0^o$, $t_S^{\hat{f}} \in T_V^o$, and $|V^F(t_S^{\hat{f}})| \leq 1$.

(2) If $\hat{f}$ is non-minimal: $t_B^{\hat{f}}, t_S^{\hat{f}} \in T^x$ where $t_B^{\hat{f}}, t_S^{\hat{f}}$ may contain only defined function symbols smaller than $\hat{f}$ in $<_x$. If $x = o$, then $|V^F(t_S^{\hat{f}})| \leq 1$ and $|V^F(t_B^{\hat{f}})| = 0$.

Additionally, $\mathcal{N}(t_B^{\hat{f}}) \subseteq \{n_1, \ldots, n_\beta\}$, $\mathcal{N}(t_S^{\hat{f}}) \subseteq \{n_1, \ldots, n_\beta\} \cup \{m, \bullet\}$ (if $\bullet \in \mathcal{N}$), and the only global variables occurring in t_B and t_S are $\overrightarrow{X}_\alpha \cup \{\bullet\}$ (if $\bullet \in V^\iota$). We define $D^x = \bigcup\{D(\hat{f}) \mid \hat{f} \in \hat{\Sigma}_x\}$.

Remark 2 We frequently write t_B instead of $t_B^{\hat{f}}$ and t_S instead of $t_S^{\hat{f}}$ when the defined symbol is clear from the context.

Definition 2 *(Closed symbol set)* Let S be a finite set of symbols in $\hat{\Sigma}_\omega \cup \hat{\Sigma}_\iota \cup \hat{\Sigma}_o$. We call S *closed* if for any $\hat{f} \in S$ all defined symbols occurring in $t_B^{\hat{f}}$ and in $t_S^{\hat{f}}$ belong to S.

Definition 3 *(Theory)* Let S be a closed set of symbols. Then the tuple $(S, \hat{f}, \mathcal{D})$ is a *theory of* $\hat{f}$ if

- $\hat{f} \in S$ and $\hat{f}$ is maximal in S,
- $\mathcal{D} = \bigcup\{D(\hat{f}) \mid \hat{f} \in S\}$.

Example 1 For $\widehat{p} \in \Sigma_\omega$, $D(\widehat{p}) = \left\{\widehat{p}(\bar{0}) = \bar{0},\ \widehat{p}(s(m)) = m\right\}$, $t_B = \bar{0}$, $t_s = m$. Let $\hat{f}, \hat{g} \in \Sigma_\omega$ s.t. $\hat{f}$ is minimal and $\hat{f} <_\omega \hat{g}$. We define $D(\hat{f})$ as

$$\hat{f}(n, \bar{0}) = t_B, \quad \hat{f}(n, s(m)) = t_S\{\bullet \leftarrow \hat{f}(n, m)\}$$

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

$$\hat{g}(n, \bar{0}) = t'_B, \quad \hat{g}(n, s(m)) = t'_S\{\bullet \leftarrow \hat{g}(n, m)\}$$

where $t'_B = \bar{0}$ and $t'_S = \hat{f}(n, \bullet)$. Then $\hat{g}$ defines $*$. In both cases $\bullet$ is any fresh parameter in $\mathcal{N}$. The corresponding theory is $\left(\{\hat{p}, \hat{f}, \hat{g}\}, \{\hat{g}\}, D(\hat{p}) \cup D(\hat{f}) \cup D(\hat{g})\right)$.

Example 2 As a second example consider $g \in \Sigma_\iota$ and $\hat{f} \in \hat{\Sigma}_\iota$. We define $D(\hat{f})$ as

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

Here, $t_B = X(0)$, $t_S = g(X(m+1), \bullet)$.

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. 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.

Definition 4 *(Rewrite systems)* Let $x \in \{\omega, \iota, o\}$, and $\hat{f} \in \Sigma_x$. Then $R(\hat{f})$ is the set of the following rewrite rules obtained from $D(\hat{f})$:

$$\hat{f}(\overrightarrow{X}_\alpha, \overrightarrow{n}_\beta, \bar{0}) \to t_B, \quad \hat{f}(\overrightarrow{X}_\alpha \overrightarrow{n}_\beta, s(m)) \to t_S\{\bullet \leftarrow \hat{f}(\overrightarrow{X}_\alpha \overrightarrow{n}_\beta, m)\},$$

$R^x = \bigcup\{R(\hat{f}) \mid \hat{f} \in \hat{\mathcal{F}}_x\}$. When a term $s \in T^x$ rewrites to t under R^x we write $s \to_x t$. for a term $s \in T^x$, such that $\mathcal{N}(s) = \emptyset$, we denote exhaustive application of R^x to s by $s\downarrow_x$, i.e. normalization of s.

Definition 4 implies that parameters ought to be replaced by numerals prior to normalization.

Definition 5 *(Parameter assignment)* A function $\sigma : \mathcal{N} \to Num$ is called a *parameter assignment*. σ is extended to $\mathcal{T}^\omega$ homomorphically:

- $\sigma(\bar{\beta}) = \bar{\beta}$ for numerals $\bar{\beta}$.
- $\sigma(s(t)) = s(\sigma(t))$
- $\sigma(\hat{f}(\overrightarrow{t_\alpha})) = \hat{f}(\sigma(\overrightarrow{t_\alpha}))\downarrow_\omega$ 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 (Definition 5) 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. Like for the terms in T^ω the evaluation is defined via a rewrite system. To evaluate a term $t \in T^\iota$ under $\sigma \in \mathcal{S}$ we have to combine $\rightarrow_\omega$ and $\rightarrow_\iota$.

Definition 6 *(Evaluation of T^ι)* Let $\sigma \in \mathcal{S}$ and $t \in T^\iota$. We define $\sigma(t)\downarrow_\iota$:

- $t = X(\overrightarrow{s_\alpha})$ for $X \in V^G$ then $\sigma(X(\overrightarrow{s_\alpha}))\downarrow_\iota = X(\overrightarrow{\sigma(s_\alpha)\downarrow_\omega})$.
- $t = f(\overrightarrow{s_\alpha})$ for $f \in \Sigma_\iota$,then $\sigma(f(\overrightarrow{s}_\alpha))\downarrow_\iota = f(\sigma(\overrightarrow{s}_\alpha)\downarrow_\iota)$.
- $t = \hat{f}(\overrightarrow{X}_\alpha, \overrightarrow{t}_{\beta+1})$ for $\hat{f} \in \hat{\Sigma}_\iota$, then $\sigma(\hat{f}(\overrightarrow{X}_\alpha, \overrightarrow{t}_{\beta+1}))\downarrow_\iota = \hat{f}(\overrightarrow{X}_\alpha, \sigma(\overrightarrow{t}_{\beta+1})\downarrow_\omega)\downarrow_\iota$.

Remark 3 Concerning global variables and normalization, we should consider the following: Let $t_1, \cdots t_\alpha, s_1, \cdots s_\alpha \in \omega$, $X, Y \in V^G$, then we say $X(t_1, \cdots t_\alpha) = Y(s_1, \cdots s_\alpha)$ iff $X = Y$ and for any parameter assignment σ we have $\sigma(t_i)\downarrow_\omega = \sigma(s_i)\downarrow_\omega$ for $1 \leq i \leq \alpha$.

Example 3 Let us consider the evaluation of the term $g(X(n), \hat{f}(X, m+1))$ with respect to the parameter assignment $\sigma(m) = 2$, $\sigma(n) = 2$, using the defining equations provided in Example 2.

$$\begin{aligned}
\sigma(g(X(n), \hat{f}(X, m+1)))\downarrow_\iota &= g(\sigma(X(n))\downarrow_\iota, \sigma(\hat{f}(X, m+1))\downarrow_\iota) \\
&= g(X(\sigma(n)\downarrow_\omega), \hat{f}(X, \sigma(m+1)\downarrow_\omega)\downarrow_\iota) \\
&= g(X(2), \hat{f}(X, 3)\downarrow_\iota) \\
&= g(X(2), g(X(3), \hat{f}(X, 2)\downarrow_\iota)) \\
&= g(X(2), g(X(3), g(X(2), \hat{f}(X, 1)\downarrow_\iota))) \\
&= g(X(2), g(X(3), g(X(2), g(X(1), \hat{f}(X, 0)\downarrow_\iota)))) \\
&= g(X(2), g(X(3), g(X(2), g(X(1), X(0)))))
\end{aligned}$$

To evaluate a term $t \in T^o$ under $\sigma \in \mathcal{S}$ we have to combine $\rightarrow_\omega$, $\rightarrow_\iota$, and $\rightarrow_o$.

Definition 7 *(Evaluation of T^o)* Let $\sigma \in \mathcal{S}$; we define $\sigma(t)\downarrow_o$ for $t \in T^o$.

- $t \in V^F$, then $\sigma(t)\downarrow_o = t$.
- $t = P(\overrightarrow{t_\alpha})$ for $P \in \Sigma_o$, then $\sigma(P(\overrightarrow{t_\alpha}))\downarrow_o = P(\overrightarrow{\sigma(t_\alpha)}\downarrow_\iota)$.
- $t = \hat{P}(\overrightarrow{X}_\alpha, \overrightarrow{t}_{\beta+1})$ for $\hat{P} \in \hat{\Sigma}_o$, then $\hat{P}(\overrightarrow{X}_\alpha, \sigma(\overrightarrow{t}_{\beta+1})\downarrow_\omega)\downarrow_o$
- $t = \neg t'$, then $\sigma(\neg t')\downarrow_o = \neg\sigma(t')\downarrow_o$.
- $t = t_1 \circ t_2$, then $\sigma(t)\downarrow_o = \sigma(t_1)\downarrow_o \circ \sigma(t_2)\downarrow_o$ for $\circ \in \{\wedge, \vee\}$.

Proposition 1 .

- *R^x is a canonical rewrite system for $x \in \{\omega, \iota, o\}$.*
- *Let $t \in T^x$ and $\sigma \in \mathcal{S}$. Then the (unique) normal form of $\sigma(t)$ under R^x, $\sigma(t)\downarrow_x$, is a member of $\mathcal{T}_0^x$.*

Proof Concerning R^ω, termination and confluence are well known, see e.g. [4]. In particular, $\bar{0}$, s and $R(\hat{\Sigma}_\omega)$ define a language for computing the set of primitive recursive functions; in particular the recursions are well founded. A formal proof of termination requires double induction on $<_\omega$ and the value of the recursion parameter. The proofs for R^ι and R^o are slightly more complex. Given the similarity of the two rule sets we will only provide formal proof for R^ι. In particular, we show that given $t \in T^\iota$ and $\sigma \in \mathcal{S}$ then $\sigma(t)\downarrow_\iota \in T_0^\iota$. We proceed according to Definition 6.

- if $t = X(\overrightarrow{s_\alpha})$ then $\sigma(\overrightarrow{s_\alpha})\downarrow_\omega = \overrightarrow{\bar{\gamma}_\alpha}$ for $\bar{\gamma}_\alpha \in Num$ and $\sigma(X(\overrightarrow{s_\alpha}))\downarrow_\iota = X(\overrightarrow{\bar{\gamma}_\alpha}) \in V^\iota$.
- if $t = f(\overrightarrow{s_\alpha})$ for $f \in \Sigma_\iota$, then $\sigma(f(\overrightarrow{s_\alpha}))\downarrow_\iota = f(\overrightarrow{\sigma(s_\alpha)\downarrow_\iota})$. By induction we may assume that $\overrightarrow{s'_\alpha} = \overrightarrow{\sigma(s_\alpha)}\downarrow_\iota \in T_0^\iota$, thus $f(s'_1, \ldots, s'_\alpha) \in T_0^\iota$.
- if $\hat{f}(\overrightarrow{X}_\alpha, \overrightarrow{t_\beta}, t_{\beta+1})$ for $\hat{f} \in \hat{\Sigma}_\iota$ and $\hat{f}$ is minimal in $<_\iota$, then we distinguish two cases

  1. $\sigma(t_{\beta+1})\downarrow_\omega = \bar{0}$. Then, $\sigma(\hat{f}(\overrightarrow{X}_\alpha, \overrightarrow{t_\beta}, t_{\beta+1}))\downarrow_\iota)) = \hat{f}(\overrightarrow{X}_\alpha, \overrightarrow{\bar{\gamma}_\beta}, \bar{0})$ for $\overrightarrow{\bar{\gamma}_i} \in Num$. According to Definition 4 $\hat{f}(\overrightarrow{X}_\alpha, \overrightarrow{\bar{\gamma}_\beta}, \bar{0})$ rewrites to $t_B \in T_0^\iota$.
  2. $\sigma(t_{\beta+1})\downarrow_\omega = p \mathbin{\bar{+}} 1$ for $p > 0$. Then $\hat{f}(\overrightarrow{X}_\alpha, \overrightarrow{\bar{\gamma}_\beta}, p \mathbin{\bar{+}} 1)$ rewrites to the term $t_S\{\bullet \leftarrow \hat{f}(\overrightarrow{X}_\alpha, \overrightarrow{\bar{\gamma}_\beta}, \bar{p})\}$ where $t_S \in T_0^\iota$. By induction on p we infer that $\hat{f}(\overrightarrow{X}_\alpha, \overrightarrow{\bar{\gamma}_\beta}, \bar{p})\downarrow_\iota \in T_0^\iota$ and so $\hat{f}(\overrightarrow{X}_\alpha, \bar{\gamma}_\beta, p \mathbin{\bar{+}} 1)$ rewrites to a term in T_0^ι.

- if $\hat{f}(\overrightarrow{X}_\alpha, \overrightarrow{t_\beta}, t_{\beta+1})$ for $\hat{f} \in \hat{\Sigma}_\iota$ and $\hat{f}$ is not minimal in $<_\iota$, then we have to add induction on $<_\iota$ with the base cases shown above.

□

Example 4 We consider the theory $(\{\hat{f}\}, \hat{f}, \mathcal{D})$ where $\mathcal{D} = \{D(\hat{f})\}$ for $D(\hat{f})$ defined below. Let $X, Y \in V^G$, $g \in \Sigma_\iota$, and n, m parameters. Assume $\hat{f}$ is defined as follows: Let $D(\hat{f})$ consist of the two equations

$$\hat{f}(X, Y, n, \bar{0}) = Y,$$
$$\hat{f}(X, Y, n, s(m)) = g(X(n, m), \hat{f}(X, Y, n, m)).$$

We evaluate $\hat{f}(X, Y, n, m)$ under σ, where $\sigma(n) = \bar{1}, \sigma(m) = \bar{2}$ and $\sigma(k) = \bar{0}$ for $k \notin \{n, m\}$.

$$\begin{aligned}\sigma(\hat{f}(X, Y, n, m))\downarrow_\iota &= \hat{f}(X, Y, \bar{1}, \bar{2})\downarrow_\iota = g(X(\bar{1}, \bar{1}), \hat{f}(X, Y, \bar{1}, \bar{1})\downarrow_\iota) \\ &= g(X(\bar{1}, \bar{1}), g(X(\bar{1}, \bar{0}), \hat{f}(X, Y, \bar{1}, \bar{0})\downarrow_\iota) = g(X(\bar{1}, \bar{1}), g(X(\bar{1}, \bar{0}), Y)).\end{aligned}$$

When we write x_1 for $X(\bar{1}, \bar{1})$ and x_2 for $X(\bar{1}, \bar{0})$ and y for Y we get the term in the common form $g(x_1, g(x_2, y))$.

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 formula. In particular, we are interested in infinite sequences of unsatisfiable formula whose refutations are finitely describable using the resolution calculus introduced later in this paper. We end this section with examples of such formulas.

Definition 8 *(Unsatisfiable schemata)* Let $F \in \mathcal{T}^o$. Then F is called *unsatisfiable* if for all $\sigma \in \mathcal{S}$ the formula $\sigma(F)\downarrow_o$ is unsatisfiable.

Example 5 Let $a \in \Sigma_\iota$, $P \in \Sigma_o$, $\hat{f}$ as in Example 2, $\hat{P}, \hat{Q} \in \hat{\Sigma}_o$ such that $\hat{P} <_o \hat{Q}$. 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:

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

$$\begin{aligned}\hat{Q}(X, Y, n, \bar{0}) &= P(\hat{f}(Y(\bar{0}), \bar{0}), Y(\bar{1})) \wedge \hat{P}(X, n) \text{ and}\\ \hat{Q}(X, Y, n, s(m)) &= P(\hat{f}(Y(\bar{0}), s(m)), Y(\bar{1})) \wedge \hat{P}(X, n).\end{aligned}$$

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

$$\begin{aligned}\sigma(\hat{Q}(X, Y, n, m))\downarrow_o &= P(\hat{f}(Y(0), 2), Y(1)) \wedge \hat{P}(X, 3)\downarrow_o)\\ &= P(\hat{f}(Y(0), 2), Y(1)) \wedge (\hat{P}(X, 2)\downarrow_o \vee\neg P(X(3), \hat{f}(a, 3)\downarrow_\iota)\\ &= P(\hat{f}(Y(0), 2), Y(1)) \wedge (\hat{P}(X, 1)\downarrow_o \vee\neg P(X(2), \hat{f}(a, 2)\downarrow_\iota) \vee\\ &\quad \neg P(X(3), \hat{f}(a, 3)\downarrow_\iota))\\ &= \ldots P(g(g(Y(0)), Y(1)) \wedge\\ &\quad (\neg P(X(0), a) \vee \neg P(X(1), g(a)) \vee \neg P(X(2), g(g(a))) \vee\\ &\quad \neg P(X(3), g(g(g(a))))).\end{aligned}$$

Note that for $\sigma(n) = \bar{\alpha}$ the number of different variables in $\sigma(\hat{Q}(X, Y, n, m))\downarrow_o$ is $\alpha + 2$; so the number of variables increases with the parameter assignments.

Example 6 Let us now consider the schematic formula representation of the inductive definition extracted from the m-successor eventually constant schema presented in Sect. 2. This requires us to define five defined predicate symbols $\hat{F}_1$, $\hat{F}_2$, $\hat{F}_3$, $\hat{F}_4$, and $\hat{F}_5$ such that $\hat{F}_5 <_o \hat{F}_4 <_o \hat{F}_3 <_o \hat{F}_2 <_o \hat{F}_1$. Furthermore, the defining equations associated with these defined predicate symbols contain the symbols $\sim, < \in \Sigma_o$, $a, f, suc \in \Sigma_\iota$, and $n, m \in \mathcal{N}$. We also require a defined function symbol $\hat{S} \in \hat{\Sigma}_\iota$. Note that in this case the sort ι is identical to ω. Using these symbols we can rewrite the inductive definition provided in Sect. 2 into the theory $(S, \hat{F}_1, \mathcal{D})$ where $S = \{\hat{F}_1, \ldots, \hat{F}_5, \hat{S}\}$ and $\mathcal{D}$ consists of the equations below ($\mathbf{X} = (X_1, X_2, X_3)$):

$$\begin{aligned}\hat{F}_1(\mathbf{X}, n, m) &= \hat{F}_2(\mathbf{X}, n, m) \wedge\ \hat{F}_3(\mathbf{X}, n, m)\\ \hat{F}_2(\mathbf{X}, n, 0) &= f(\hat{S}(X_1(n, 0), 0)) \sim n\ \vee\ f(X_1(n, 0)) < n\\ \hat{F}_2(\mathbf{X}, n, s(m)) &= (f(\hat{S}(X_1(n, s(m)), s(m))) \sim n\ \vee\ f(X_1(n, s(m))) < n) \wedge\\ &\quad \hat{F}_2(\mathbf{X}, n, m)\\ \hat{F}_3(\mathbf{X}, 0, m) &= \hat{F}_5(\mathbf{X}, 0, m)\ \wedge\ f(a) \nless 0\\ \hat{F}_3(\mathbf{X}, s(n), m) &= (\hat{F}_5(\mathbf{X}, s(n), m)\ \wedge (\hat{F}_4(\mathbf{X}, n, m))\ \wedge \hat{F}_3(\mathbf{X}, n, m)\\ \hat{F}_4(\mathbf{X}, n, 0) &= f(\hat{S}(X_2(n, 0), 0)) \nless s(n)\ \vee\ f(\hat{S}(X_2(n, 0), 0)) \sim n\ \vee\\ &\quad f(X_2(n, 0)) < n\\ \hat{F}_4(\mathbf{X}, n, s(m)) &= f(\hat{S}(X_2(n, s(m)), s(m))) \nless s(n)\ \vee\\ &\quad f(\hat{S}(X_2(n, s(m)), s(m))) \sim n \vee f(X_2(n, s(m))) < n\ \wedge\\ &\quad \hat{F}_4(\mathbf{X}, n, m)\\ \hat{F}_5(\mathbf{X}, n, 0) &= f(\hat{S}(X_3(n), 0)) \nsim n\\ \hat{F}_5(\mathbf{X}, n, s(m)) &= f(\hat{S}(X_3(n), s(m))) \nsim n\ \vee\ \hat{F}_5(\mathbf{X}, n, m)\\ \hat{S}(Z, 0) &= Z\\ \hat{S}(Z, s(n)) &= suc(\hat{S}(Z, n))\end{aligned}$$

In dealing with term schemata we have to consider schematic substitutions, particularly when we are interested in unification. Below we develop some formal tools to describe such schemata. Note that for two term schemata to be unifiable, they have to be unifiable for all parameter assignments. 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 shown below in Example 7, there are term schemata (which are defined without using global variables) that are unifiable for some, but not all parameter assignments.

Example 7 Let us consider $\hat{f}$,$\hat{f}_1$, and $\hat{g}$ with the defining equations

$$\begin{aligned} \hat{f}(x,0) &= h(a,a),\ \hat{f}(x,s(n)) &&= h(x,\hat{f}(x,n)) \\ \hat{f}_1(x,y,0) &= h(a,a),\ \hat{f}_1(x,y,s(n)) &&= h(x,\hat{f}(y,n)) \\ \hat{g}(x,y,0) &= h(a,a),\ \hat{g}(x,y,s(n)) &&= h(\hat{g}(x,y,n),y) \end{aligned}$$

Note that $\hat{f}_1 > \hat{f}$. Consider the parameter assignment $\sigma = \{n \rightarrow 2\}$ and the evaluation of $\hat{f}_1(x,y,n)$:

$$\begin{aligned} \hat{f}_1(x,y,n)\downarrow_\sigma &= \hat{f}_1(x,y,2)\downarrow \\ &= h(x,\hat{f}(y,1)\downarrow) = h(x,h(y,\hat{f}(x,0)\downarrow)) = h(x,h(y,h(a,a)). \end{aligned}$$

We can define unification problems such as

$$\hat{f}(x,n) \overset{?}{=} \hat{g}(y,y,n)$$

Consider $\sigma_0 = \{n \rightarrow 0\}$ and $\sigma_1 = \{n \rightarrow 1\}$. Then, the unification problem evaluates to

$$\begin{aligned} \hat{f}(x,n)\downarrow_{\sigma_0} \overset{?}{=} \hat{g}(y,y,n)\downarrow_{\sigma_0} &\Rightarrow h(a,a) \overset{?}{=} h(a,a) \\ \hat{f}(x,n)\downarrow_{\sigma_1} \overset{?}{=} \hat{g}(y,y,n)\downarrow_{\sigma_1} &\Rightarrow h(x,h(a,a)) \overset{?}{=} h(h(a,a),y), \end{aligned}$$

both of which are unifiable. However, for $\sigma_2 = \{n \rightarrow 2\}$ the unification problem evaluates to

$$h(x,h(x,h(a,a))) \overset{?}{=} h(h(h(a,a),y),y).$$

After two steps unification fails due to occurs check.
On the other hand,

$$\hat{f}_1(x,y,s(n)) \overset{?}{=} \hat{g}(z,z,s(n))$$

is a unifiable unification problem. The evaluation for $\sigma_2 = \{n \rightarrow 2\}$ is

$$h(x,h(y,h(y,h(a,a)))) = h(h(h(h(a,a),z),z),z).$$

A unifier for this problem is $\theta =$

$$\{x \leftarrow h(h(h(a,a),h(y,h(y,h(a,a)))),h(y,h(y,h(a,a)))),z \leftarrow h(y,h(y,h(a,a)))\}.$$

The substitution schema is

$$\hat{\vartheta}(n) = \{x \leftarrow \hat{g}(\hat{f}(y,n),\hat{f}(y,n),n),\ z \leftarrow \hat{f}(y,n)\}.$$

As term schemata that are defined without the use of global variables repeat a finite set of variables arbitrarily often, in many cases the unification problem of term schemata will result in occurrence check failure. We can tackle this problem by using global variables. Usually, we do not desire all variables occurrences to be the same nor do we desire them to all be different. These extreme cases can be described through quantification. Let P be a one-place predicate symbol, then $P(\hat{f}(x,n))$ can be interpreted as

$$\forall x P(h(x, h(x, \ldots, h(x, h(a,a))\ldots))), \text{ or as}$$
$$\forall x_1, \ldots x_n P(h(x_1, h(x_2, \ldots, h(x_n, h(a,a))\ldots))).$$

The use of global variables allows for the syntactic description of properties of the quantifier prefix. Moreover, it reduces unwanted occurrence check failure. The domain of a unifier of term schemata, that are constructed using global variables, is by construction dependent on the numeric parameter. These kind of unifiers are called s-unifiers. Before introducing s-unification formally, we need some preliminaries.
The class T_0^ω which represents the free algebra based on s and $\bar{0}$ is not very expressive while T^ω is too strong (many properties are undecidable). For our proof analysis in Sect. 6 we need a slight extension of T_0^ω; besides the successor, we add the predecessor in order to define recursive calls. For this reason we extend our class T_0^ω by adding the defined function symbol $\hat{p}$ as defined in Example 1.

Definition 9 (T_1^ω) Let $\hat{p} \in \hat{\Sigma}_\omega$ and $D(\hat{p})$ as in Example 1. The class T_1^ω is defined inductively as follows.

- $\bar{0} \in T_1^\omega$,
- $\mathcal{N} \subseteq T_1^\omega$,
- if $t \in T_1^\omega$ then $s(t) \in T_1^\omega$,
- if $t \in T_1^\omega$ then $\hat{p}(t) \in T_1^\omega$.

Definition 10 *(Essentially distinct)* Let $\overrightarrow{s}_1 = (s_1, \ldots, s_\alpha)$ for $s_1, \ldots, s_\alpha \in T_1^\omega$ and $\overrightarrow{s}_2 = (s'_1, \ldots, s'_\beta)$ for $s'_1, \ldots, s'_\beta \in T_1^\omega$. $\overrightarrow{s}_1$ and $\overrightarrow{s}_2$ are called *essentially distinct* if either $\alpha \neq \beta$ or for all $\sigma \in \mathcal{S}$ there exists an $i \in \{1, \ldots, \alpha\}$ such that $\sigma(s_i)\downarrow_\omega \neq \sigma(s'_i)\downarrow_\omega$.

Proposition 2 *Let $\alpha \geq 0$, $\overrightarrow{s}_1 = (s_1, \ldots, s_\alpha)$ for $s_1, \ldots, s_\alpha \in T_1^\omega$ and $\overrightarrow{s}_2 = (s'_1, \ldots, s'_\alpha)$ for $s'_1, \ldots, s'_\alpha \in T_1^\omega$, and $\Gamma = \{s_1 \overset{?}{=} s'_1, \cdots, s_\alpha \overset{?}{=} s'_\alpha\}$. Then Γ is unifiable over T_1^ω (i.e. in the theory $(\{\hat{p}\}, \{\hat{p}\}, \{D(\hat{p})\})$ over T_1^ω) iff $\overrightarrow{s}_1$ and $\overrightarrow{s}_2$ are not essentially distinct.*

Proposition 3 *It is decidable whether $\overrightarrow{s}$, $\overrightarrow{t}$ are essentially distinct for term tuples $\overrightarrow{s}$, $\overrightarrow{t}$ in T_1^ω.*

Proof If the arity of $\overrightarrow{s}$ and $\overrightarrow{t}$ is different the problem is trivial. Therefore we consider terms of the form

$$\overrightarrow{s} = (s_1, \ldots, s_\alpha), \ \overrightarrow{t} = (t_1, \ldots, t_\alpha).$$

By Proposition 2 $\overrightarrow{s}$, $\overrightarrow{t}$ are not essentially distinct iff

$$\Gamma : \{\{s_1 \overset{?}{=} t_1, \cdots, s_\alpha \overset{?}{=} t_\alpha\}$$

is solvable over T_1^ω. We present an algorithm for deciding solvability of such a system Γ.
Let t be a term in T_1^ω. Then t is either of the form $f_1 \cdots f_\beta n$ for $n \in \mathcal{N}$ or $f_1 \cdots f_\beta \bar{0}$ where $f_1, \ldots, f_\beta \in \{s, \hat{p}\}$. To each such term and $\sigma \in \mathcal{S}$ we assign an arithmetic expression $\pi(\sigma, t)$:

- If $t = f_1 \cdots f_\beta \bar{0}$ then $\pi(\sigma, t) = \alpha$ for $f_1 \cdots f_\beta \bar{0}\downarrow_\omega = \bar{\alpha}$.
- If $t = f_1 \cdots f_\beta n$ we define
  - $\nu_s(t) =$ number of occurrences of s in t,

– $\nu_{\hat{p}}(t) =$ number of occurrences of $\hat{p}$ in t.

$$\begin{aligned}\pi(\sigma, t) &= n + \nu_s(t) - \nu_{\hat{p}}(t) \text{ for } \sigma(n) \geq \overline{\nu_{\hat{p}}(t)},\\ &= \alpha_i \text{ for } \bar{\alpha_i} = f_1 \cdots f_\beta \bar{i}\downarrow_\omega \text{ and } \sigma(n) = \bar{i}, i < \nu_{\hat{p}}(t).\end{aligned}$$

Now let $\mathcal{T}(n)$ be all terms of the form $f_1 \cdots f_\beta n$ in Γ. Select the t in $\mathcal{T}(n)$ where $\nu_{\hat{p}}(t)$ is maximal and define $r(n) = \nu_{\hat{p}(t)}$. If $n_1, \ldots, n_\gamma$ are all the variables in Γ we obtain numbers $r(n_1), \ldots, r(n_\gamma)$. For all terms t of the form $f_1 \cdots f_\beta n$ we define now

$$\begin{aligned}\pi(\sigma, t) &= n + \nu_s(t) - \nu_{\hat{p}}(t) \text{ for } \sigma(n) \geq \overline{r(n)},\\ &= \alpha_i \text{ for } \bar{\alpha_i} = f_1 \cdots f_\beta \bar{i} \text{ and } \sigma(n) = \bar{i}, i < r(n).\end{aligned}$$

Now consider the valid formula

$$\left(\bigvee_{i=0}^{r(n_1)-1} n_1 = i \vee n_1 \geq r(n_1)\right) \wedge \cdots \wedge \left(\bigvee_{i=0}^{r(n_\gamma)-1} n_\gamma = i \vee n_\gamma \geq r(n_\gamma)\right)$$

and transform it into an equivalent DNF F. Then every conjunct C in F defines a condition on σ such that the $\pi(\sigma, s_i), \pi(\sigma, t_i)$ are uniquely defined and every C defines a system

$$\mathcal{E}(C, \sigma) = \{\pi(\sigma, s_1) \overset{?}{=} \pi(\sigma, t_1), \ldots, \pi(\sigma, s_\alpha) \overset{?}{=} \pi(\sigma, t_\alpha)\}.$$

The solvability of $\mathcal{E}(C, \sigma)$ is easy to check as all equations are of the form $m \circ i = n \star j$, $m \circ i = j$ or $i = j$ for $\circ, \star \in \{+, -\}$, m, n integer variables and $i, j \in \mathbb{N}$. But Γ is solvable iff all the $\mathcal{E}(C, \sigma)$ are solvable. Hence the decision algorithm consists in checking all equational systems $\mathcal{E}(C, \sigma)$ for solvability. □

Example 8 Let $\overrightarrow{s} = (\hat{p}sn, m, n)$, $\overrightarrow{t} = (\hat{p}\hat{p}n, n, sk)$. For m, k we get $\pi(\sigma, m) = m, \pi(\sigma, sk) = k + 1$. We have two terms "ending" with n, namely $\hat{p}sn$ and $\hat{p}\hat{p}n$. Here we get

$$\begin{aligned}&\pi(\sigma, \hat{p}sn) = n \text{ for } \sigma(n) \geq \bar{1},\ \pi(\sigma, \hat{p}s\bar{n}) = 0 \text{ for } \sigma(n) = \bar{0},\\ &\pi(\sigma, \hat{p}\hat{p}n) = 0 \text{ for } \sigma(n) < \bar{2},\ \pi(\sigma, \hat{p}\hat{p}n) = n - 2 \text{ for } \sigma(n) \geq \bar{2}.\end{aligned}$$

We obtain $r(n) = 2$ and obtain the formula $n = 0 \vee n = 1 \vee n \geq 2$ which is already in DNF. The corresponding equation systems are

$$\begin{aligned}\mathcal{E}_1 &= \{0 = 0,\ m = 0,\ 0 = k + 1\},\\ \mathcal{E}_2 &= \{1 = 0,\ m = 1,\ 1 = k + 1\},\\ \mathcal{E}_3 &= \{n = n - 2,\ m = n,\ n = k + 1\}.\end{aligned}$$

All equation systems are unsolvable and thus $\overrightarrow{s}$, $\overrightarrow{t}$ are essentially distinct. It is easy to see that the first equation system is solvable if we change the term $\overrightarrow{t}$ to $\overrightarrow{t'} = (\hat{p}\hat{p}n, n, k)$. So $\overrightarrow{s}$, $\overrightarrow{t'}$ are not essentially distinct.

Definition 11 *(s-substitution)* Let Θ be a finite set of pairs $(X(\overrightarrow{s}_\alpha), t)$ where $X(\overrightarrow{s}_\alpha) \in T_V^\iota$, $\overrightarrow{s}_\alpha$ a tuple of terms in T_1^ω and $t \in T^\iota$. Note that the global variables occurring in Θ need not be of the same type. Θ is called an *s-substitution* if for all $(X(\overrightarrow{s}_\alpha), t)$, $(Y(\overrightarrow{s'}_\alpha), t') \in \Theta$ either $X \neq Y$ or the tuples $\overrightarrow{s}_\alpha$ and $\overrightarrow{s'}_\alpha$ are essentially distinct. For $\sigma \in \mathcal{S}$ we define $\Theta[\sigma] = \{X(\sigma(\overrightarrow{s}_\alpha)\downarrow_\omega) \leftarrow t\sigma\downarrow_\iota \mid (X(\overrightarrow{s}_\alpha), t) \in \Theta\}$.
We define $dom(\Theta) = \{X(\overrightarrow{s}_\alpha) \mid (X(\overrightarrow{s}_\alpha), t) \in \Theta\}$ and $rg(\Theta) = \{t \mid (X(\overrightarrow{s}_\alpha), t) \in \Theta\}$.

Proposition 4 *For all $\sigma \in \mathcal{S}$ and every s-substitution Θ, $\Theta[\sigma]$ is a (first-order) substitution.*

Proof It is enough to show that for all $(X(\overrightarrow{s}_{\alpha}), t), (Y(\overrightarrow{s'}_{\alpha'}), t') \in \Theta$ $X(\sigma(\overrightarrow{s}_{\alpha})\downarrow_{\omega}) \neq Y(\sigma(\overrightarrow{s'}_{\alpha'})\downarrow_{\omega})$ and $\sigma(t)\downarrow_{\iota}, \sigma(t')\downarrow_{\iota} \in T_0^{\iota}$ (follows from Proposition 1), for all $\sigma \in \mathcal{S}$. If $X \neq Y$ this is obvious; if $X = Y$ then, by definition of Θ, $\overrightarrow{s}_{\alpha}$ and $\overrightarrow{s'}_{\alpha}$ are essentially distinct and so for each $\sigma \in \mathcal{S}$ we have $X(\sigma(\overrightarrow{s}_{\alpha})\downarrow_{\omega}) \neq X(\sigma(\overrightarrow{s'}_{\alpha'})\downarrow_{\omega})$. Thus $\Theta[\sigma]$ is indeed a substitution as for $X(\overrightarrow{s}_{\alpha}) \in T_V^{\iota}$ $X(\sigma(\overrightarrow{s}_{\alpha})) \in V^{\iota}$. □

Example 9 The following expression is an s-substitution

$$\Theta = \{(X(n,m), \hat{S}(Y(m), n)), (X(s(n), m), \hat{S}(Y(m), s(n))), (X(0,0), Y(0))\}.$$

for $\hat{S}$ as in Example 6.

The application of an s-substitution Θ to terms in T^{ι} is defined inductively on the complexity of term definitions as usual.

Definition 12 Let Θ be an s-substitution and σ a parameter assignment. We define $t\Theta[\sigma]$ for terms $t \in T^{\iota}$:

- if t is a constants of type ι, then $t\Theta[\sigma] = t$,
- if $t = X(\overrightarrow{s}_{\alpha})$ and $(X(\overrightarrow{s'}_{\alpha}), t') \in \Theta$ such that $X(\sigma(\overrightarrow{s'}_{\alpha})) = X(\sigma(\overrightarrow{s}_{\alpha}))$, then $X(\overrightarrow{s}_{\alpha})\Theta[\sigma] = \sigma(t')\downarrow_{\iota}$, otherwise $X(\overrightarrow{s}_{\alpha})\Theta[\sigma] = X(\sigma(\overrightarrow{s}_{\alpha}))$;
- if $f \in \mathcal{F}_{\iota}$, $f\colon \iota^{\alpha} \to \iota$, $s_1, \ldots, s_{\alpha} \in T^{\iota}$ then $f(s_1, \ldots, s_{\alpha})\Theta[\sigma] = f(s_1\Theta[\sigma], \ldots, s_{\alpha}\Theta[\sigma])$,
- if $\hat{f} \in \hat{\Sigma}_{\iota}$, $\hat{f}\colon \tau(\gamma(1), \ldots, \gamma(\alpha_1)) \times \omega^{\beta+1} \to \iota$, then
$$\hat{f}(\overrightarrow{X}_{\alpha_1}, t_1, \ldots, t_{\beta+1})\Theta[\sigma] = \hat{f}(\overrightarrow{X}_{\alpha_1}, \sigma(t_1)\downarrow_{\omega}, \ldots, \sigma(t_{\beta+1})\downarrow_{\omega})\downarrow_{\iota} \Theta[\sigma].$$

Example 10 Let us consider the following defined function symbol:

$$\begin{aligned}\hat{g}(X, n, \bar{0}) &= X(n, 0),\\ \hat{g}(X, n, s(m)) &= g(X(n, m), \hat{g}(X, s(n), m)).\end{aligned}$$

and the parameter assignment $\sigma = \{n \leftarrow 0, m \leftarrow s(0)\}$. Then the evaluation of the term $\hat{g}(X, n, m)$ by the s-substitution Θ from Example 9 proceeds as follows:

$$\begin{aligned}\hat{g}(X, n, m)\Theta[\sigma] &= \hat{g}(X, \sigma(n)\downarrow_{\omega}, \sigma(m)\downarrow_{\omega})\downarrow_{\omega} \Theta[\sigma]\\ &= \hat{g}(X, 0, s(0))\downarrow_{\omega} \Theta[\sigma]\\ &= g(X(0, s(0)), \hat{g}(X, s(0), 0)\downarrow_{\omega})\Theta[\sigma]\\ &= g(X(0, s(0)), X(s(0), 0))\Theta[\sigma] = g(Y(s(0)), X(s(0), 0)).\end{aligned}$$

where

$$\Theta[\sigma] = \{(X(0, s(0)), Y(s(0))), (X(s(0), s(0)), suc(Y(s(0)))), (X(0,0), Y(0))\}.$$

for $\hat{S}$ as in Example 6.

The composition of s-substitutions is not trivial as, in general, there is no uniform representation of composition under varying parameter assignments.

Example 11 Let $\Theta_1 = \{(X_1(n), f(X_1(n))\}$ and $\Theta_2 = \{(X_1(0), g(a))\}$. Then, for $\sigma \in \mathcal{S}$ s.t. $\sigma(n) = 0$ we get

$$\Theta_1[\sigma] \circ \Theta_2[\sigma] = \{X_1(0) \leftarrow f(X_1(0))\} \circ \{X_1(0) \leftarrow g(a)\} = \{X_1(0) \leftarrow f(g(a))\}.$$

On the other hand, for $\sigma' \in \mathcal{S}$ with $\sigma'(n) = 1$ we obtain

$$\Theta_1[\sigma'] \circ \Theta_2[\sigma'] = \{X_1(1) \leftarrow f(X_1(1))\} \circ \{X_1(0) \leftarrow g(a)\} = \{X_1(1) \leftarrow f(X_1(1)), X_1(0) \leftarrow g(a)\}.$$

Or take $\Theta_1' = \{(X_1(n), X_2(n))\}$ and $\Theta_2' = \{(X_2(m), X_1(m))\}$.
Let $\sigma(n) = \sigma(m) = 0$ and $\sigma'(n) = 0, \sigma'(m) = 1$. Then

$$\Theta_1'[\sigma] \circ \Theta_2'[\sigma] = \{X_2(0) \leftarrow X_1(0)\},$$
$$\Theta_1'[\sigma'] \circ \Theta_2'[\sigma'] = \{X_1(0) \leftarrow X_2(0), X_2(1) \leftarrow X_1(1)\}.$$

The examples above suggest the following restrictions on s-substitutions with respect to composition. The first definition ensures that domain and range are variable-disjoint.

Definition 13 Let Θ be an s-substitution. Θ is called *normal* if for all $\sigma \in \mathcal{S}$ $dom(\Theta[\sigma]) \cap V^\iota(rg(\Theta[\sigma])) = \emptyset$.

Example 12 The s-substitution in Example 9 is normal. The substitutions Θ_1' and Θ_2' in Example 11 are normal. Θ_1 in Example 11 is not normal.

Proposition 5 *It is decidable whether a given s-substitution is normal.*

Proof Let Θ be an s-substitution. We search for equal global variables in $dom(\Theta)$ and in $rg(\Theta)$; if there are none then Θ is trivially normal. So let $X \in V^G(dom(\Theta)) \cap V^G(rg(\Theta))$. For every $X(\overrightarrow{s}_\alpha) \in dom(\Theta)$ and for every $X(\overrightarrow{t}_\alpha)$ occurring in $rg(\Theta)$ we test unifiability of the arguments in the sense of Proposition 2. Θ is normal iff for no pair $X(\overrightarrow{s}_\alpha$ and $X(\overrightarrow{t}_\alpha)$ are the arguments unifiable in the sense of Proposition 2. □

Example 11 shows also that normal s-substitutions cannot always be composed to an s-substitution; thus we need an additional condition.

Definition 14 Let Θ_1, Θ_2 be normal s-substitutions. (Θ_1, Θ_2) is called *composable* if for all $\sigma \in \mathcal{S}$

1. $dom(\Theta_1[\sigma]) \cap dom(\Theta_2(\sigma)) = \emptyset$,
2. $dom(\Theta_1[\sigma]) \cap V^\iota(rg(\Theta_2[\sigma])) = \emptyset$.

Proposition 6 *It is decidable whether (Θ_1, Θ_2) is composable for two normal s-substitutions Θ_1, Θ_2.*

Proof Like in Proposition 5 we use Proposition 2 to test unifiability of arguments for variables $X(\overrightarrow{s}), X(\overrightarrow{t})$ occurring in the sets under consideration. □

Definition 15 Let Θ_1, Θ_2 be normal s-substitutions and (Θ_1, Θ_2) composable. Assume that

$$\Theta_1 = \{(X_1(\overrightarrow{s_1}), t_1), \ldots, (X_\alpha(\overrightarrow{s_\alpha}), t_\alpha)\}, \quad \Theta_2 = \{(Y_1(\overrightarrow{w_1}), r_1), \ldots, (Y_\beta(\overrightarrow{w_\beta}), r_\beta)\}.$$

Then the composition $\Theta_1 \star \Theta_2$ is defined as

$$\{(X_1(\overrightarrow{s_1}), t_1\Theta_2), \ldots, (X_\alpha(\overrightarrow{s_\alpha}), t_\alpha\Theta_2), (Y_1(\overrightarrow{w_1}), r_1), \ldots, (Y_\beta(\overrightarrow{w_\beta}), r_\beta)\}.$$

The following proposition shows that $\Theta_1{\star}\Theta_2$ really represents composition.

Proposition 7 *Let Θ_1, Θ_2 be normal s-substitutions and (Θ_1, Θ_2) be composable then for all $\sigma \in \mathcal{S}$ $(\Theta_1{\star}\Theta_2)[\sigma] = \Theta_1[\sigma] \circ \Theta_2[\sigma]$.*

Proof Let

$$\Theta_1 = \{(X_1(\overrightarrow{s_1}), t_1), \ldots, (X_\alpha(\overrightarrow{s_\alpha}), t_\alpha)\}, \quad \Theta_2 = \{(Y_1(\overrightarrow{w_1}), r_1), \ldots, (Y_\beta(\overrightarrow{w_\beta}), r_\beta)\}.$$

Then $\Theta_1{\star}\Theta_2$ is defined as

$$\{(X_1(\overrightarrow{s_1}), t_1\Theta_2), \ldots, (X_\alpha(\overrightarrow{s_\alpha}), t_\alpha\Theta_2), (Y_1(\overrightarrow{w_1}), r_1), \ldots, (Y_\beta(\overrightarrow{w_\beta}), r_\beta)\}.$$

We write x_i for $X_i(\sigma(\overrightarrow{s_i})))$ and y_j for $Y_j(\sigma(\overrightarrow{w_j}))$, θ_1 for $\Theta_1[\sigma]$ and θ_2 for $\Theta_2[\sigma]$. Moreover let $t'_i = \sigma(t_i){\downarrow_\iota}$, $r'_j = \sigma(r_j){\downarrow_\iota}$. Then

$$\theta_1 = \{x_1 \leftarrow t'_1, \ldots, x_\alpha \leftarrow t'_\alpha\}, \quad \theta_2 = \{(y_1 \leftarrow r'_1, \ldots, y_{\alpha'} \leftarrow r'_\beta\}.$$

As (Θ_1, Θ_2) is composable we have

1. $\{x_1, \ldots, x_\alpha\} \cap \{y_1, \ldots, y_\beta\} = \emptyset$, and
2. $\{x_1, \ldots, x_\alpha\} \cap V^\iota(\{r'_1, \ldots, r'_\beta\}) = \emptyset$.

So $\theta_1\theta_2 = \{x_1 \leftarrow t'_1, \ldots, x_\alpha \leftarrow t'_\alpha)\}\theta_2 = \{x_1 \leftarrow t'_1\theta_2, \ldots, x_\alpha \leftarrow t'_\alpha\theta_2\} \cup \theta_2$. The last substitution is just $(\Theta_1{\star}\Theta_2)[\sigma]$. □

Proposition 8 *Let Θ_1, Θ_2 be normal s-substitutions and (Θ_1, Θ_2) composable. Then $\Theta_1{\star}\Theta_2$ is normal.*

Proof Like in the proof of Proposition 7 let $\Theta_1[\sigma] = \theta_1$, $\Theta_2[\sigma] = \theta_2$. We have to show that $dom(\theta_1\theta_2) \cap V^\iota(rg(\theta_1\theta_2) = \emptyset$. We have

$$\theta_1\theta_2 = \{x_1 \leftarrow t'_1\theta_2, \ldots, x_\alpha \leftarrow t'_\alpha\theta_2\} \cup \theta_2.$$

As θ_1 is normal we have $V^\iota(t'_i) \cap \{x_1, \ldots, x_\alpha\} = \emptyset$ for $i = 1, \ldots, \alpha$. By definition of composability $rg(\theta_2) \cap \{x_1, \ldots, x_\alpha\} = \emptyset$, and therefore

$$V^\iota(\{t'_1\theta_2, \ldots, t'_\alpha\theta_2\}) \cap \{x_1, \ldots, x_\alpha\} = \emptyset.$$

So $\{x_1 \leftarrow t'_1\theta_2, \ldots, x_\alpha \leftarrow t'_\alpha\theta_2\}$ is normal. As also Θ_2 is normal we have $dom(\theta_2) \cap V^\iota(rg(\theta_2) = \emptyset$. Hence we obtain $dom(\theta_1\theta_2) \cap V^\iota(rg(\theta_1\theta_2)) = \emptyset$. □

Definition 16 *(s-unifier)* Let $t_1, t_2 \in T^\iota$. An s-substitution Θ is called an s-unifier of t_1, t_2 if for all $\sigma \in \mathcal{S}$ $(t_1\sigma{\downarrow_\iota})\Theta[\sigma] = (t_2\sigma{\downarrow_\iota})\Theta[\sigma]$. We refer to t_1, t_2 as s-unifiable if there exists an s-unifier of t_1, t_2. s-unifiability can be extended to more than two terms and to formula schemata (to be defined below) in an obvious way.

Example 13 Consider the following theory $\left(\{\hat{f}, \hat{g}\}, \{\hat{f}\}, D(\hat{f}) \cup D(\hat{g})\right)$ where

$$D(\hat{f}) = \{\ \hat{f}(X, \mathbf{0}) \ = \ h(a, X(0))\ ,\ \hat{f}(X, s(n)) \ = \ h(X(s(n)), \hat{f}(X, n))\}$$

and

$$D(\hat{g}) = \{\ \hat{g}(X, \mathbf{0}) \ = \ h(X(0), a)\ ,\ \hat{g}(X, s(n)) \ = \ h(\hat{g}(X, n), X(s(n)))\}$$

Using these schemata we can define the unification problem

$$\{\hat{f}(X, s(n)), \hat{g}(X, s(n))\}$$

which has as a unification schema $\hat{\Theta} : \left\{ X(n) \leftarrow \hat{h}(n) \right\}$ where $\hat{h}(n)$ is as follows:

$$D(\hat{h}) = \{\hat{h}(\mathbf{0}) = X(0), \hat{h}(s(n)) = h(\hat{h}(n), \hat{h}(n))\}.$$

$\hat{\Theta}(n)$ is an s-unifier within the extended theory

$$\left(\{\hat{f}, \hat{g}, \hat{h}\}, \{\hat{f}\}, D(\hat{f}) \cup D(\hat{g}) \cup D(\hat{h}) \right).$$

Definition 17 An s-unifier Θ of t_1, t_2 is called *restricted* to $\{t_1, t_2\}$ if $T_V^{\iota}(\Theta) \subseteq T_V^{\iota}(\{t_1, t_2\})$.

Remark 4 It is easy to see that for any s-unifier Θ of $\{t_1, t_2\}$ there exists an s-unifier Θ' of $\{t_1, t_2\}$ which is *restricted* to $\{t_1, t_2\}$.

Most general unification is defined modulo the set of parameter substitutions $\mathcal{S}$.

Definition 18 A restricted s-unifier Θ of $\{t_1, t_2\}$ is a *most general unifier* if for all parameter substitutions $\sigma \in \mathcal{S}$, $\Theta\,[\sigma]$ is a most general unifier of $\{\sigma(t_1){\downarrow_\iota}, \sigma(t_2){\downarrow_\iota}\}$.

Remark 5 For example, the s-unifier from Example 13 is a most general unifier. Note that it is not clear if a most general unifier always exists. We do not have a decision procedure for the unification problem, not even for restricted classes.

4 The Resolution Calculus

The basis of our calculus for refuting formula schemata is a calculus RPL_0 for quantifier-free formulas, which combines dynamic normalization rules (a la Andrews, see [1]) with the resolution rule. In contrast to [1] we do not restrict the resolution rule to atomic formulas. We denote as PL_0 the set of quantifier-free formulas in predicate logic; for simplicity we omit $\rightarrow$ and represent it by $\neg$ and $\vee$ in the usual way. *Sequents* are objects of the form $\Gamma \vdash \Delta$ where Γ and Δ are multisets of formulas in PL_0.

Definition 19 (RPL_0) The axioms of RPL_0 are sequents $\vdash F$ for $F \in \mathrm{PL}_0$. The rules are the elimination rules for the connectives

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

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

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

the introduction rules for the connectives

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B}\ \wedge_{r}^{+} \quad \frac{A, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta}\ \wedge_{l_1}^{+} \quad \frac{B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta}\ \wedge_{l_2}^{+}$$

$$\frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta, A \vee B}\ \vee_{r_1}^{+} \quad \frac{\Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \vee B}\ \vee_{r_2}^{+} \quad \frac{A, \Gamma \vdash \Delta \quad B, \Gamma \vdash \Delta}{A \vee B, \Gamma \vdash \Delta}\ \vee_{l}^{+}$$

$$\frac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A}\ \neg_r^+ \quad \frac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta}\ \neg_l^+$$

the resolution rule

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

ϑ is an m.g.u. of $\{A_1, \ldots, A_k, B_1, \ldots, B_l\}$, $V(\{A_1, \ldots, A_k\}) \cap V(\{B_1, \ldots, B_l\}) = \emptyset$.

We will extend RPL_0 by rules handling schematic formula definitions. But we have to consider another aspect as well: 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. To this aim we added introduction rules for the connectives, which gives us the potential to derive more complex formulas. Note that our aim is to use the calculi in an interactive way and not fully automatic, which justifies this process of “anti-refinement”.

Proposition 9 RPL_0 *is sound and refutationally complete, i.e.*

(1) *all rules in* RPL_0 *are sound and*
(2) *for any unsatisfiable formula* $\forall F$ *and* $F \in \mathrm{PL}_0$ *there exists a* RPL_0*-derivation of* $\vdash$ *from axioms of the form* $\vdash F\vartheta$ *where* ϑ *is a renaming of* $V(F)$.

Proof (1) is trivial: if $\mathcal{M}$ is a model of the premise(s) of a rule then $\mathcal{M}$ is also a model of the conclusion.
For proving (2) we first derive the standard clause set $\mathcal{C}$ of F. Therefore, we apply the rules of RPL_0 to $\vdash F$, decomposing F into its subformulas, until we cannot apply any rule other than the resolution rule res. The last subformula obtained in this way is atomic and hence a clause. The standard clause set $\mathcal{C}$ of F is comprised of the clauses obtained in this way. As $\forall F$ is unsatisfiable, its standard clause set is refutable by resolution. Thus, we apply res to the clauses and obtain $\vdash$. The whole derivation lies in RPL_0. □

In extending RPL_0 to a schematic calculus we have to replace unification by s-unification. Formally we have to define how s-substitutions are extended to formula schemata and sequent schemata.

Definition 20 Let Θ be an s-substitution. We define $F\Theta$ for all $F \in \mathcal{T}^o$ which do not contain formula variables.

- Let $P(t_1, \ldots, t_\alpha) \in T^o$ and $P \in \mathcal{P}$. Then $P(t_1, \ldots, t_\alpha)\Theta = P(t_1\Theta, \ldots, t_\alpha\Theta)$.
- Let $\hat{P} \in \hat{\mathcal{P}}$ and $\hat{P}(X_1, \ldots, X_\alpha, t_1, \ldots, t_{\beta+1}) \in T^o$, then

$$\hat{P}(X_1, \ldots, X_\alpha, t_1, \ldots, t_{\beta+1})\Theta = \hat{P}(X_1, \ldots, X_\alpha, t_1\Theta, \ldots, t_{\beta+1}\Theta).$$

- $(\neg F)\Theta = \neg F\Theta$.
- If $F_1, F_2 \in \mathcal{T}^o$ then

$$(F_1 \wedge F_2)\Theta = F_1\Theta \wedge F_2\Theta, \quad (F_1 \vee F_2)\Theta = F_1\Theta \vee F_2\Theta.$$

Let $S\colon A_1, \ldots, A_\alpha \vdash B_1, \ldots, B_\beta$ be a sequent schema. Then

$$S\Theta = A_1\Theta, \ldots, A_\alpha\Theta \vdash B_1\Theta, \ldots, B_\beta\Theta.$$

In the resolution rule we have to ensure that the sets of variables in $\{A_1, \dots, A_k\}$ and $\{B_1, \dots, B_l\}$ are pairwise disjoint. We need a corresponding concept of disjointness for the schematic case.

Definition 21 *(Essentially disjoint)* Let $\mathcal{A}, \mathcal{B}$ be finite sets of schematic variables in T^{ι}_{V}. $\mathcal{A}$ and $\mathcal{B}$ are called *essentially disjoint* if for all $\sigma \in \mathcal{S}$ $\mathcal{A}[\sigma] \cap \mathcal{B}[\sigma] = \emptyset$.

Definition 22 *(*RPL^{Ψ}_0*)* Let $\Psi \colon (S, \hat{Q}, \mathcal{D})$ be a theory of the schematic predicate symbol $\hat{Q}$ then, for all schematic predicate symbols $\hat{P} \in S$ for

$$D(\hat{P}) = \{\hat{P}(\mathbf{Y}, \mathbf{n}, 0) = t_B, \quad \hat{P}(\mathbf{Y}, \mathbf{n}, s(m)) = t_S\{\bullet \leftarrow \hat{P}(\mathbf{Y}, \mathbf{n}, m)\},$$

elimination of defined symbols

$$\frac{\Gamma \vdash \Delta, \hat{P}(\mathbf{Y}, \mathbf{n}, 0)}{\Gamma \vdash \Delta, t_B}\ B\hat{P}r \qquad \frac{\Gamma \vdash \Delta, \hat{P}(\mathbf{Y}, \mathbf{n}, s(m))}{\Gamma \vdash \Delta, t_S\{\bullet \leftarrow \hat{P}(\mathbf{Y}, \mathbf{n}, m)\}}\ S\hat{P}r$$

$$\frac{\hat{P}(\mathbf{Y}, \mathbf{n}, 0), \Gamma \vdash \Delta}{t_B, \Gamma \vdash \Delta}\ B\hat{P}l \qquad \frac{\hat{P}(\mathbf{Y}, \mathbf{n}, s(m)), \Gamma \vdash \Delta}{t_S\{\bullet \leftarrow \hat{P}(\mathbf{Y}, \mathbf{n}, m)\}, \Gamma \vdash \Delta}\ S\hat{P}l$$

introduction of defined symbols

$$\frac{\Gamma \vdash \Delta, t_B}{\Gamma \vdash \Delta, \hat{P}(\mathbf{Y}, \mathbf{n}, 0)}\ B\hat{P}r^{+} \qquad \frac{\Gamma \vdash \Delta, t_S\{\bullet \leftarrow \hat{P}(\mathbf{Y}, \mathbf{n}, m)\}}{\Gamma \vdash \Delta, \hat{P}(\mathbf{Y}, \mathbf{n}, s(m))}\ S\hat{P}r^{+}$$

$$\frac{t_B, \Gamma \vdash \Delta}{\hat{P}(\mathbf{Y}, \mathbf{n}, 0), \Gamma \vdash \Delta}\ B\hat{P}l^{+} \qquad \frac{t_S\{\bullet \leftarrow \hat{P}(\mathbf{Y}, \mathbf{n}, m)\}, \Gamma \vdash \Delta}{\hat{P}(\mathbf{Y}, \mathbf{n}, s(m)), \Gamma \vdash \Delta}\ S\hat{P}l^{+}$$

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

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

The refutational completeness of RPL^{Ψ}_0 is not an issue as already RPL_0 is refutationally complete for PL_0 formulas [3,14]. Note that this is not the case any more if parameters occur in formulas. Indeed, due to the usual theoretical limitations, the logic is not semi-decidable for schematic formulas [2]. RPL^{Ψ}_0 is sound if the defining equations are considered.

Proposition 10 *Let the sequent S be derivable in* RPL^{Ψ}_0 *for $\Psi = (S', \hat{P}, \mathcal{D})$. Then $\mathcal{D} \models S$.*

Proof The introduction and elimination rules for defined predicate symbols are sound with respect to $\mathcal{D}$; also the resolution rule (involving s-unification) is sound with respect to $\mathcal{D}$. □

Definition 23 An RPL^{Ψ}_0 derivation ϱ is called a *cut-derivation* if the s-unifiers of all resolution rules are empty.

Remark 6 A cut-derivation is an RPL^{Ψ}_0 derivation with only propositional rules. Such a derivation can be obtained by combining all unifiers to a global unifier.

In computing global unifiers we have to apply s-substitutions to proofs. However, not every s-substitution applied to a RPL_0^Ψ derivation results in a RPL_0^Ψ derivation again. Just assume that an s-unifier in a resolution is of the form $(X_1(s), X_2(s'))$; if $\Theta = \{(X_1(s), a), (X_2(s'), b)\}$ for different constant symbols a, b then $X_1(s)\Theta$ and $X_2(s')\Theta$ are no longer unifiable and the resolution is blocked.

Definition 24 Let ρ be a derivation in RPL_0^Ψ which does not contain the resolution rule; then for any s-substitution Θ $\rho\Theta$ is the derivation in which every sequent occurrence S is replaced by $S\Theta$. We say that Θ is admissible for ρ. Now let $\rho =$

$$\dfrac{\overset{(\rho_1)}{\Gamma \vdash \Delta, A_1, \ldots, A_\alpha} \quad \overset{(\rho_2)}{B_1, \ldots, B_\beta, \Pi \vdash \Lambda}}{\Gamma\Theta', \Pi\Theta' \vdash \Delta\Theta', \Lambda\Theta'}\ res$$

where Θ' is an s-unifier of $\{A_1, \ldots, A_\alpha, B_1, \ldots, B_\beta\}$. Let us assume that Θ is admissible for ρ_1 and ρ_2. We define that Θ is admissible for ρ if the set

$$U\colon \{A_1\Theta, \ldots, A_\alpha\Theta, B_1\Theta, \ldots, B_\beta\Theta\}$$

is s-unifiable. If Θ^* is an s-unifier of U then we can define $\rho\Theta$ as

$$\dfrac{\overset{(\rho_1\Theta)}{\Gamma\Theta \vdash \Delta\Theta, A_1\Theta, \ldots, A_\alpha\Theta} \quad \overset{(\rho_2\Theta)}{B_1\Theta, \ldots, B_\beta\Theta, \Pi\Theta \vdash \Lambda\Theta}}{\Gamma\Theta\Theta^*, \Pi\Theta\Theta^* \vdash \Delta\Theta\Theta^*, \Lambda\Theta\Theta^*}\ res$$

Definition 25 Let ϱ be an RPL_0^Ψ derivation and Θ be an s-substitution which is admissible for ϱ. Θ is called a *global unifier* for ϱ if $\varrho\Theta$ is a cut-derivation.

In order to compute global unifiers we need RPL_0^Ψ derivations in some kind of "normal form". Below we define two necessary restrictions on derivations.

Definition 26 An RPL_0^Ψ derivation ϱ is called *normal* if all s-unifiers of resolution rules in ϱ are normal and restricted.

Remark 7 Note that, in case of s-unifiability, we can always find normal and restricted s-unifiers; thus the definition above does not really restrict the derivations, it only requires some renamings.

Definition 27 An RPL_0^Ψ derivation ϱ is called *regular* if for all subderivations ϱ' of ϱ of the form

$$\dfrac{\overset{(\varrho_1')}{\Gamma \vdash \Delta} \quad \overset{(\varrho_2')}{\Pi \vdash \Lambda}}{\Gamma', \Pi' \vdash \Delta', \Lambda'}\ \chi$$

we have $V^G(\varrho_1') \cap V^G(\varrho_2') = \emptyset$.

Note that the condition $V^G(\varrho_1') \cap V^G(\varrho_2') = \emptyset$ in Definition 27 guarantees that, for all parameter assignments σ, $\varrho_1'[\sigma]$ and $\varrho_2'[\sigma]$ are variable-disjoint.
We write $\varrho' \leq_{ss} \varrho$ if there exists an s-substitution Θ s.t. $\varrho'\Theta = \varrho$.

Proposition 11 *Let ϱ be a normal* RPL_0^Ψ *derivation. Then there exists a* RPL_0^Ψ *derivation ϱ' s.t. $\varrho' \leq_{ss} \varrho$ and ϱ' is normal and regular.*

Proof By renaming of variables in subproofs and in s-unifiers. □

Proposition 12 *Let ϱ be a normal and regular* RPL_0^Ψ *derivation. Then there exists a global s-unifier Θ for ϱ which is normal and $V^G(\Theta) \subseteq V^G(\varrho)$.*

Proof By induction on the number of inferences in ϱ.
Induction base: ϱ is an axiom. $\emptyset$ is a global s-unifier which trivially fulfils the properties.
For the induction step we distinguish two cases.

- The last rule in ϱ is unary. Then ϱ is of the form

$$\dfrac{\begin{matrix}(\varrho')\\ \Gamma' \vdash \Delta'\end{matrix}}{\Gamma \vdash \Delta}\ \xi$$

  By induction hypothesis there exists a global substitution Θ' which is a global unifier for ϱ' s.t. Θ' is normal and $V^G(\Theta') \subseteq V^G(\varrho')$. We define $\Theta = \Theta'$. Then, trivially, Θ is normal and a global unifier of ϱ. Moreover, by definition of the unary rules in RPL_0^Ψ, we have $V^G(\varrho') = V^G(\varrho)$ and so $V^G(\Theta) \subseteq V^G(\varrho)$.
- ϱ is of the form

$$\dfrac{\begin{matrix}(\varrho_1)\\ \Gamma \vdash \Delta, A_1, \ldots, A_\alpha\end{matrix} \quad \begin{matrix}(\varrho_2)\\ B_1, \ldots, B_\beta, \Pi \vdash \Lambda\end{matrix}}{\Gamma\Theta, \Pi\Theta \vdash \Delta\Theta, \Lambda\Theta}\ res(\Theta)$$

  As ϱ is a normal RPL_0^Ψ derivation the unifier Θ is normal. By regularity of ϱ we have $V^G(\varrho_1) \cap V^G(\varrho_2) = \emptyset$.
  By induction hypothesis there exist global normal unifiers Θ_1, Θ_2 for ϱ_1 and ϱ_2 s.t. $V^G(\Theta_1) \subseteq V^G(\varrho_1)$ and $V^G(\Theta_2) \subseteq V^G(\varrho_2)$. By $V^G(\varrho_1) \cap V^G(\varrho_2) = \emptyset$ we also have $V^G(\Theta_1) \cap V^G(\Theta_2) = \emptyset$.
  We show now that (Θ_1, Θ) and (Θ_2, Θ) are composable. As Θ_1 is normal we have for all $\sigma \in \mathcal{S}$

$$V^\iota(\{A_1, \ldots, A_\alpha\}[\sigma]) \cap dom(\Theta_1[\sigma]) = \emptyset.$$

  Similarly we obtain

$$V^\iota(\{B_1, \ldots, B_\beta\}[\sigma]) \cap dom(\Theta_2[\sigma]) = \emptyset.$$

  As Θ is normal and restricted we have for all $\sigma \in \mathcal{S}$

$$V^\iota(\Theta[\sigma]) \subseteq V^\iota(\sigma\{A_1, \ldots, A_\alpha, B_1, \ldots, B_\beta\}{\downarrow_o}).$$

  Therefore (Θ_1, Θ) and (Θ_2, Θ) are both composable. As $\Theta_1, \Theta_2, \Theta$ are normal so are $\Theta_1\star\Theta$ and $\Theta_2\star\Theta$. As Θ_1, Θ_2 are essentially disjoint we can define

$$\Theta(\varrho) = \Theta_1\star\Theta \cup \Theta_2\star\Theta.$$

  $\Theta(\varrho)$ is a normal s-substitution and $V^G(\Theta(\varrho)) \subseteq V^G(\varrho)$.
  $\Theta(\varrho)$ is also a global unifier of ϱ. Indeed, $\varrho_1\Theta(\varrho) =$

$$\begin{matrix}(\varrho_1\Theta(\varrho))\\ \Gamma\Theta \vdash \Delta\Theta, A_1\Theta, \ldots, A_1\Theta\end{matrix}$$

  and $\varrho_2\Theta(\varrho) =$

$$\begin{matrix}(\varrho_2\Theta(\varrho))\\ A_1\Theta, \ldots, A_1\Theta, \Pi\Theta \vdash \Lambda\Theta\end{matrix}$$

So we obtain the derivation

$$\dfrac{\varrho_1\Theta(\varrho) \quad \varrho_2\Theta(\varrho)}{\Gamma\Theta, \Pi\Theta \vdash \Delta\Theta, \Lambda\Theta}\ cut$$

which is an instance of ϱ and a cut derivation (note that every instance of a cut derivation is a cut derivation as well).

- ϱ is of the form

$$\dfrac{\begin{array}{c}(\varrho_1)\\ \Gamma \vdash \Delta\end{array} \qquad \begin{array}{c}(\varrho_2)\\ \Pi \vdash \Lambda\end{array}}{\Gamma', \Pi' \vdash \Delta', \Lambda'}\ \chi$$

where χ is a binary rule different from resolution.
As ϱ is a normal RPL_0^{Ψ} derivation all occurring s-unifiers in ϱ_1 and ϱ_2 are normal. By regularity of ϱ we have that $V^G(\varrho_1) \cap V^G(\varrho_2) = \emptyset$.
By induction hypothesis there exist global normal unifiers Θ_1, Θ_2 for ϱ_1 and ϱ_2 s.t. $V^G(\Theta_1) \subseteq V^G(\varrho_1)$ and $V^G(\Theta_2) \subseteq V^G(\varrho_2)$. By $V^G(\varrho_1) \cap V^G(\varrho_2) = \emptyset$ we also have $V^G(\Theta_1) \cap V^G(\Theta_2) = \emptyset$. Moreover, there is no overlap between the domain variables of the unifiers Θ_1 and Θ_2, i.e. $dom(\Theta_1[\sigma]) \cap dom(\Theta_2(\sigma)) = \emptyset$ for all $\sigma \in \mathcal{S}$. Therefore, we can define $\Theta = \Theta_1 \cup \Theta_2$, which is obviously a global s-unifier of ϱ. Furthermore, $V^G(\Theta) = V^G(\Theta_1) \cup V^G(\Theta_2)$, therefore $V^G(\Theta) \subseteq V^G(\varrho_1) \cup V^G(\varrho_2)$ and by definition of binary introduction rules in RPL_0^{Ψ}, we have $V^G(\Theta) \subseteq V^G(\varrho)$. □

Example 14 We provide a simple RPL_0^{Ψ} refutation using the schematic formula constructed in Example 6. We will only cover the RPL_0^{Ψ} derivation of the base case and wait for the introduction of proof schemata to provide a full refutation. We abbreviate X_1, X_2, X_3 by $\mathbf{X}$.

$$\cfrac{\cfrac{\cfrac{\cfrac{\begin{array}{c}\left(\delta_0^1, (0,0)\right)\\ \vdash \hat{F}_1(\mathbf{X},0,0)\end{array}}{\vdash \hat{F}_2(\mathbf{X},0,0) \wedge \hat{F}_3(\mathbf{X},0,0)}\ S\hat{F}_1r}{\vdash \hat{F}_2(\mathbf{X},0,0)}\ \wedge : r}{\vdash f(X_1(0,0)) < 0 \vee f(X_1(0,0)) \sim 0}\ B\hat{F}_2r}{\vdash f(X_1(0,0)) < 0, f(X_1(0,0)) \sim 0}\ \vee : r$$

$$(2)$$

$$\cfrac{(2) \qquad \cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\begin{array}{c}\left(\delta_0^2, (0,0)\right)\\ \vdash \hat{F}_1(\mathbf{X},0,0)\end{array}}{\vdash \hat{F}_2(\mathbf{X},0,0) \wedge \hat{F}_3(\mathbf{X},0,0)}\ S\hat{F}_1r}{\vdash \hat{F}_3(\mathbf{X},0,0)}\ \wedge : r}{\vdash \hat{F}_5(\mathbf{X},0,0) \wedge f(a) \not< 0}\ S\hat{F}_3r}{\vdash \hat{F}_5(\mathbf{X},0,0)}}{\vdash f(\hat{S}(X_3(0,0),0)) \not\sim 0}\ B\hat{F}_5r}{\vdash f(X_3(0,0)) \not\sim 0}\ B\hat{S}r}{f(X_3(0,0)) \sim 0 \vdash}\ \neg : r}{\vdash f(X_1(0,0)) < 0}\ res\left(\left\{ X_3(0,0) \leftarrow X_1(0,0) \right\}\right)$$

$$(1)$$

$$\dfrac{\overset{(1)}{\vdash f(X_1(0,0)) < 0} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{\left(\delta_0^3, (0,0)\right)}{\vdash \hat{F}_1(\mathbf{X},0,0)}}{\vdash \hat{F}_2(\mathbf{X},0,0) \wedge \hat{F}_3(\mathbf{X},0,0)}\; S\hat{F}_1 r}{\vdash \hat{F}_3(\mathbf{X},0,0)}\; \wedge : r}{\vdash \hat{F}_5(\mathbf{X},0,0) \wedge f(a) \not< 0}\; B\hat{F}_3 r}{\vdash f(a) \not< 0}\; \wedge : r_2}{f(a) < 0 \vdash}\; \neg : r}{\underset{(\delta_0, 0, 0)}{\vdash}}\; res\,(\{X_1(0,0) \leftarrow a\})$$

The labels $(\delta_0^1, (0,0)), \ldots, (\delta_0^3, (0,0)$ can be ignored for the moment. They become important when the derivation above becomes part of a proof schema to be defined in Sect. 6.

RPL_0^{Ψ}-derivations can be evaluated under parameter assignments. Let φ be an RPL_0^{Ψ}-derivation and σ a parameter assignment, then $\varphi\downarrow_\sigma$ denotes the RPL_0-derivation defined by φ under σ. Note that, if the parameters are all replaced by numerals then occurrences and introductions of defined symbols can be treated as instances of definition rules (see Section 7.3 [7]). Removal of defined symbols can be treated as definition rule elimination (a cosmetic change) and thus $\varphi\downarrow_\sigma$ is indeed an RPL_0-derivation.

5 Point Transition Systems

For our specification of schematic proofs we will use complex call structures beyond primitive recursion. To characterize such complex recursion types we develop an abstract framework. Consider, e.g., the primitive recursive definitions of $+$ and $*$ (we write p for $+$, t for $*$ and s for successor):

$$\begin{aligned} t(n,0) &= 0,\\ t(n,m+1) &= p(t(n,m),n),\\ p(n,0) &= n,\\ p(n,m+1) &= s(p(n,m)). \end{aligned}$$

In defining t we assume that p has been defined "before", while p is defined via recursion and the successor s (which is a base symbol). In fact we can order the function symbols t and p by defining $p < t$, where $<$ is irreflexive and transitive. The relation $<$ prevents that both $p < t$ and $t < p$ holds, as then we would get $p < p$ contradicting irreflexivity. So primitive recursion, being based on orderings of function symbols, excludes the use of mutual recursion. However, mutual recursion is a very powerful specification principle which, even when equivalent primitive recursive specifications exist, may provide simpler and more elegant representations.

Example 15 We assume that $+$ is already defined (e.g. as p above). We define three functions f, g, h, where f is defined via f and g and g via f and h making an ordering of the symbols f, g impossible.

$$\begin{aligned} f(n+1,m+1) &= f(n,m+1) + g(n+1,m),\\ g(n+1,m) &= f(n,m) + h(n+1,n,m), \end{aligned}$$

$$f(0, m+1) = 1, \qquad f(n+1, 0) = 0,$$
$$f(0,0) = 0,$$
$$g(0, m) = m,$$
$$h(0, n, m) = m+1, \qquad h(k+1, m, n) = h(k, m, n) + 1.$$

f and g are not defined via primitive recursion and therefore it is not so simple to prove that f and g are indeed total and thus recursive. That the above type of mutual recursion is terminating is based on $(n+1, m+1) > (n+1, m)$, $(n+1, m+1) > (n+1, m)$ and $(n+1, m) > (n, m)$ where $>$ is the lexicographic tuple ordering. That the definition is also well defined (we obtain a value for all $(n, m) \in \mathbb{N}^2$) follows from the following partitions of $\mathbb{N}^2$:

$\{(n, m) \mid n > 0, m > 0\}$, $\{(n, m) \mid n = 0, m > 0\}$, $\{(n, m) \mid n > 0, m = 0\}$,
$\{(n, m) \mid n = 0, m = 0\}$ for f, and
$\{(n, m) \mid n > 0\}$, $\{(n, m) \mid n = 0\}$ for g,
$\{(k, m, n) \mid k > 0\}$, $\{(k, m, n) \mid k = 0\}$ for h.

In this section we abstract from the computation of functions and even of values of any kind. Instead our aim is to focus on the underlying recursion itself. For Example 15 above we choose a symbol δ_1 for f, δ_2 for g, δ_3 for h and $\delta_4, \ldots, \delta_8$ as termination symbols. Then the recursion of the example can be represented by the following conditional reduction rules:

$$\begin{aligned}
(\delta_1, (n, m)) &\to \{(\delta_1, (n-1, m)), (\delta_2, (n, m-1))\} \text{ for } m > 0 \wedge n > 0,\\
(\delta_1, (n, m)) &\to \{(\delta_4, (n, m))\} \text{ for } m > 0 \wedge n = 0,\\
(\delta_1, (n, m)) &\to \{(\delta_5, (n, m))\} \text{ for } m = 0 \wedge n > 0,\\
(\delta_1, (n, m)) &\to \{(\delta_6, (n, m))\} \text{ for } m = 0 \wedge n = 0,\\
(\delta_2, (n, m)) &\to \{(\delta_1, (n-1, m)), (\delta_3, (n, n-1, m))\} \text{ for } n > 0,\\
(\delta_2, (n, m)) &\to \{(\delta_7, (n, m))\} \text{ for } n = 0,\\
(\delta_3, (k, n, m)) &\to \{(\delta_3, (k-1, n, m))\} \text{ for } k > 0,\\
(\delta_3, (k, n, m)) &\to \{(\delta_8, (k, n, m)\} \text{ for } k = 0.
\end{aligned}$$

We call such a system a *point transition system*; the formal definitions are given below.

Definition 28 *(Point)* A *point* is an element of $\mathcal{P}^\alpha$ for $\mathcal{P} \subseteq T^\omega$ and some $\alpha \geq 1$.

Definition 29 *(Labeled point)* Let Δ be an infinite set of symbols called *labels*. A *labeled point* is a pair (δ, p) where p is a point and $\delta \in \Delta$.

Definition 30 *(Condition)* An *atomic condition* is either $\top$, $\bot$ or of the form $s < t$, $s > t$, $s = t$, $s \neq t$ for $s, t \in T_0^\omega$. Let ATC be the set of all atomic conditions. We define the set COND of all conditions inductively:

- ATC $\subseteq$ COND,
- if $C \in$ COND then $\neg C \in$ COND,
- if $C_1, C_2 \in$ COND then $C_1 \wedge C_2 \in$ COND and $C_1 \vee C_2 \in$ COND.

Example 16 $k = 0$ and $m < n$ are atomic conditions. $k = 0 \wedge m < n$ and $(k = 0 \wedge m < n) \vee m = n$ are conditions.

Definition 31 *(Semantics of conditions)* For every $C \in$ COND and $\sigma \in \mathcal{S}$ we define $\sigma[C] \in \{\top, \bot\}$:

- $\sigma[\top] = \top,\ \sigma[\bot] = \bot$.
- $\sigma[s = t] = \top$ if $s\!\downarrow_\sigma = t\!\downarrow_\sigma$, and $\bot$ otherwise.
- $\sigma[s < t] = \top$ if $s\!\downarrow_\sigma < t\!\downarrow_\sigma$, and $\bot$ otherwise.
- $\sigma[s > t] = \top$ if $s\!\downarrow_\sigma > t\!\downarrow_\sigma$, and $\bot$ otherwise.
- $\sigma[s \neq t] = \top$ iff $\sigma[s = t] = \bot$.
- $\sigma[C_1 \wedge C_2] = \top$ if $\sigma[C_1] = \top$ and $\sigma[C_2] = \top$, and $\bot$ otherwise.
- $\sigma[C_1 \vee C_2] = \top$ if $\sigma[C_1] = \top$ or $\sigma[C_2] = \top$, and $\bot$ otherwise.

A condition C is called *valid* if for all $\sigma \in \mathcal{S}$: $\sigma[C] = \top$. C is called *unsatisfiable* if for all $\sigma \in \mathcal{S}$: $\sigma[C] = \bot$.

Definition 32 *(Point transition)* A *point transition* is an expression of the form $(\delta, p) \to \mathcal{P}\colon C$ where (δ, p) is a labeled point and $\mathcal{P}$ is a nonempty finite set of labeled points and C is a condition. We define functions l, r, c on point transitions t as follows: if $t = (\delta, p) \to \mathcal{P}\colon C$ then $l(t) = (\delta, p),\ r(t) = \mathcal{P},\ c(t) = C$.

Definition 33 *(Point transition cluster)* A finite set of point transitions $\mathbf{P}$ is called a *point transition cluster* if for all $\delta \in \Delta$ and points p, q such that (δ, p) and (δ, q) occur in $\mathbf{P}$ (as $l(t)$ or in $r(t)$ for a $t \in \mathbf{P}$) there exists an $\alpha \geq 1$ such that $p, q \in \mathcal{P}^\alpha$. That means for all δ occurring in $\mathbf{P}$ (for this set we write $\Delta(\mathbf{P})$) the corresponding points are all of the same arity; this arity is denoted by $A(\delta)$.

Example 17 Let

$$\begin{aligned}\mathbf{P}_1 = \{&(\delta, (m, n)) \to \{(\delta, (m, s(n))), (\delta', (m, m, m))\}\colon 0 < m,\\ &(\delta, (k, l)) \to (\delta', (k, l, l))\colon 0 < k \wedge l = 0\}.\end{aligned}$$

Then $\mathbf{P}_1$ is a point transition cluster. Here we have $A(\delta) = 2$, $A(\delta') = 3$. The following set $\mathbf{P}_2$ of point transitions for

$$\begin{aligned}\mathbf{P}_2 = &(\delta, (m, n)) \to \{(\delta, (m, m, n))\}\colon 0 < m,\\ &(\delta, (k, l)) \to \{(\delta', (k, l, l))\}\colon 0 < k \wedge l = 0\}.\end{aligned}$$

is not a point transition cluster as $A(\delta)$ cannot be defined consistently.

Point transition systems are restricted forms of point transition clusters where the labeled points and the conditions are subject to further restrictions.

Definition 34 *(Partition)* Let $\mathcal{C} = \{C_1, \ldots, C_\alpha\}$ be a set of conditions. $\mathcal{C}$ is called a *partition* if $C_1 \vee \cdots \vee C_\alpha$ is valid and for all $i, j \in \{1, \ldots, \alpha\}$ and $i \neq j$ $C_i \wedge C_j$ is unsatisfiable.

Definition 35 *(Regular point transition)* Let $\mathbf{p}\colon (\delta, p) \to \mathcal{P}\colon C$ be a point transition in a point transition cluster and $p \in \mathcal{P}^\alpha$. $\mathbf{p}$ is called *regular* if

- $p = (n_1, \ldots, n_\alpha)$ for distinct parameters $n_1, \ldots, n_\alpha$,
- for all $(\delta', q) \in \mathcal{P}$ $V(q) \subseteq \{n_1, \ldots, n_\alpha\}$,
- $V(C) \subseteq \{n_1, \ldots, n_\alpha\}$.

Definition 36 *(Point transition system)* Let $\mathbf{P}$ be a point transition cluster $\delta_0 \in \Delta(\mathbf{P})$. Then the tuple $\mathbf{P}^\star\colon (\delta_0, \Delta^*, \Delta_e, \mathbf{P})$ is called a *point transition system* if the following conditions are fulfilled:

1. $\delta_0 \in \Delta^*$,
2. $\Delta^* = \Delta(\mathbf{P})$

3. All point transitions in $\mathbf{P}$ are regular.
4. Let $t_1, t_2 \in \mathbf{P}$ with $l(t_1) = (\delta, p)$ and $l(t_2) = (\delta, q)$; then $p = q$. That means every left-hand side of a point transition with δ is related to a unique point p; we call p the *source* of δ.
5. Let $\mathbf{P}(\delta)$ be the set $\{(\delta, p) \rightarrow \mathcal{P}_1 : C_1, \ldots, (\delta, p) \rightarrow \mathcal{P}_\alpha : C_\alpha\}$ consisting of all $t \in \mathbf{P}$ with $l(t) = (\delta, p)$ (for the source p of δ). Then, in case $\mathbf{P}(\delta) \neq \emptyset$, $\{C_1, \ldots, C_\alpha\}$ is a partition.
6. $\mathbf{P}(\delta_0) \neq \emptyset$.
7. $\mathbf{P}(\delta) = \emptyset$ for $\delta \in \Delta_e$.

The label δ_0 is called the *start label* of $\mathbf{P}^\star$, the δ in Δ_e are called *end labels*.

Every point transition system defines a kind of computation for a given $\sigma \in \mathcal{S}$.

Definition 37 (σ-trees) Let $\mathbf{P}^\star : (\delta_0, \Delta^*, \Delta_e, \mathbf{P})$ be a point transition system, $\sigma \in \mathcal{S}$ and (δ, p) be a labeled point in $\mathcal{P}$. We define the *σ-tree* on a labeled point (δ, p) inductively:

- $T(\mathbf{P}, (\delta, p), \sigma, 0)$ is the tree consisting only of a root node labeled with $(\delta, \sigma(p)\downarrow)$.
- Let us assume that $T(\mathbf{P}, (\delta, p), \sigma, \alpha)$ is already defined. For every leaf ν of T_α labeled with (δ', q) we check whether $\mathcal{P}(\delta') = \emptyset$; if this is the case then ν remains a leaf node. If $\mathcal{P}(\delta') \neq \emptyset$ then

  $$\mathcal{P}(\delta') = \{(\delta', p') \rightarrow \mathcal{P}_1 : C_1, \ldots, (\delta', p') \rightarrow \mathcal{P}_\beta : C_\beta\}$$

  Let us assume that $q \in \mathcal{P}^\alpha$ and q is a tuple of numerals . Then, as $\mathcal{P}(\delta')$ is a set of regular point transitions, $p' = (n_1, \ldots, n_\alpha)$ for distinct parameters $n_1, \ldots, n_\alpha$. As q is a tuple of numerals $(q_1, \ldots, q_\alpha)$ $q = \sigma'(p')\downarrow$ for a parameter assignment σ' where $\sigma'(n_i) = q_i$ for $i = 1, \ldots, \alpha$. As all point transitions in $\mathcal{P}(\delta')$ are regular we have $V(C_i) \subseteq \{n_1, \ldots, n_\alpha\}$ for $i = 1, \ldots, \beta$. Now, $\{C_1, \ldots, C_\beta\}$ is a partition, thus there exists exactly one $i \in \{1, \ldots, \beta\}$ such that $\sigma'[C_i] = \top$. Let $\mathcal{P}_i = \{(\delta^i_1, q^i_1), \ldots, (\delta^i_{r(i)}, q^i_{r(i)})\}$. Then we create new nodes $\mu_1, \ldots, \mu_{r(i)}$ labeled with the labeled points

  $$(\delta^i_1, \sigma'(q^i_1)\downarrow), \ldots, (\delta^i_{r(i)}, \sigma'(q^i_{r(i)})\downarrow) \text{ and add the edges}$$
  $$(\nu, \mu_1), \ldots, (\nu, \mu_{r(i)}) \text{ to } T(\alpha)$$

  By the regularity of $\mathbf{P}^\star$ the points $\sigma'(q^i_j)\downarrow$ are all tuples of numerals. We call the tree obtained from $T(\mathbf{P}, (\delta, p), \sigma, \alpha)$ via the procedure above $T(\mathbf{P}, (\delta, p), \sigma, \alpha + 1)$.

Let $T(\mathbf{P}, (\delta, p), \sigma, \alpha) = (V_\alpha, E_\alpha)$ for all $\alpha \in \mathbb{N}$. Note that for all α we have $V_\alpha \subseteq V_{\alpha+1}$ and $E_\alpha \subseteq E_{\alpha+1}$. So, when we define $V_\infty = \bigcup_{\alpha \in \mathbb{N}} V_\alpha$ and $E_\infty = \bigcup_{\alpha \in \mathbb{N}} E_\alpha$, then (V_∞, E_∞) is a tree; we call this tree the *σ-tree on* (δ, p) and denote it by $T(\mathbf{P}, (\delta, p), \sigma)$. Note that the tree $T(\mathbf{P}, (\delta, p), \sigma)$ may be infinite. It is finite if there exists an α such that $T(\alpha) = T(\alpha + 1)$ The result of a computation of $\mathbf{P}^\star$ with respect to σ is defined by $T_{\mathbf{P}^\star}[\sigma] = T(\mathbf{P}, (\delta_0, p_0), \sigma)$.

We have seen that for every point transition system $\mathbf{P}^\star$ and $\sigma \in \mathcal{S}$ $T_{\mathbf{P}^\star}[\sigma]$ is a tree with nodes labeled by labeled points describing the computation of $\mathbf{P}^\star$ on input σ. However, $T_{\mathbf{P}^\star}[\sigma]$ may be infinite what means that the computation is nonterminating.

Example 18 Let $\mathbf{P}^\star = (\delta, \{\delta, \delta'\}, \{\delta'\}, \mathbf{P})$ for

$$\mathbf{P} = \{(\delta, n) \rightarrow \{(\delta, s(n))\} : n > 0,\ (\delta, n) \rightarrow \{(\delta', n)\} : n = 0\}.$$

$\mathbf{P}^\star$ is a point transformation system. The source of δ is n and $\{0 < n, n = 0\}$ is a partition. For σ with $\sigma(n) = \bar{0}$ we obtain $T_{\mathbf{P}^\star}[\sigma] =$

$$\frac{(\delta', \bar{0})}{(\delta, \bar{0})}$$

But for every σ with $\sigma(n) = \bar{\alpha} > \bar{0}$ we obtain the infinite tree

$$\frac{\dfrac{\dots}{(\delta, \overline{\alpha + 1})}}{(\delta, \bar{\alpha})}$$

Indeed, for every $i \in \mathbb{N}$, $T(i + 1) \neq T(i)$ for $T(i)$ as defined in Definition 37. So the computation of $\mathbf{P}^\star$ on σ is nonterminating.

Definition 38 Let $\mathcal{P}^\star$ be a point transition system. $\mathcal{P}^\star$ is called *terminating* if $T_{\mathcal{P}^\star}[\sigma]$ is finite for all $\sigma \in \mathcal{S}$.

To ensure termination of a point transition system we have to define a well-founded order on the set of labeled points. The ordering $<_P$ defined below will be used in Sect. 6 for recursive definitions of proofs.

Definition 39 *($<_P$)* We define an order on the set of all labeled points (δ, p) where $\delta \in \Delta_0$ for a finite subset Δ_0 of Δ. We partition Δ_0 in two disjoint subsets Δ_1, Δ_2 where Δ_2 will be reserved for mutual recursion and Δ_1 stands for primitive recursive order types. We define an irreflexive, transitive relation $<_{\Delta_0}$ on Δ_0 such that $\delta <_{\Delta_0} \delta'$ for all $\delta \in \Delta_1$ and $\delta' \in \Delta_2$. Let $\alpha > 0$ and $\mathcal{P}_0^\alpha$ be the subspace of the point space $\mathcal{P}^\alpha$ consisting of α-tuples of numerals only. For $\mathcal{P}_0^\alpha$ we have a well-founded total order $<_\alpha$. We extend $<_\alpha$ to $\mathcal{P}^\alpha$ by defining

$$\text{for } p, q \in \mathcal{P}^\alpha\colon\ p <_\alpha q \text{ if for all } \sigma \in \mathcal{S}\colon \sigma(p){\downarrow} <_\alpha \sigma(q){\downarrow}\ .$$

We are now extending the orderings to an order $<_P$ of labeled points over Δ_0. Let $(\delta, p), (\delta', q)$ two such points. We define $(\delta, p) <_P (\delta', q)$ if

(P1) $\delta, \delta' \in \Delta_2$, $A(\delta) = A(\delta')$ and $p <_\alpha q$ for $\alpha = A(\delta)$ or
(P2) $\delta, \delta' \in \Delta_2$ and $A(\delta') < A(\delta)$ or
(P3) $\delta \in \Delta_1, \delta' \in \Delta_2$ or
(P4) $\delta, \delta' \in \Delta_1$ and $\delta <_{\Delta_0} \delta'$ or
(P5) $\delta, \delta' \in \Delta_1, \delta = \delta'$ and $p <_\alpha q$ for $\alpha = A(\delta)$.

It is easy to see that the conditions (P1)–(P5) exclude each other. Note that (P3) implies $\delta <_{\Delta_0} \delta'$.

The relation $<_P$ is stable under parameter assignments and is well founded.

Proposition 13 *Let $(\delta, p) <_P (\delta', q)$ and $\sigma \in \mathcal{S}$. Then $(\delta, \sigma(p){\downarrow}) <_P (\delta', \sigma(q){\downarrow})$.*

Proof For (P1), (P5) this follows directly from the definition of $<_\alpha$ on $\mathcal{P}^\alpha$. (P2)–(P4) are invariant under parameter assignments. □

Proposition 14 *$<_P$ is well-founded.*

Proof We show first that $<_P$ is irreflexive and transitive. That $<_P$ is irreflexive can be immediately read of from the conditions (P1)–(P5). It remains to show transitivity. To this aim we have to check all combinations of (P1)–(P5). Let us assume $(\delta, p) <_P (\delta', q)$ and $(\delta', q) <_P (\delta'', r)$.

(P1)–(P1): Here we have $A(\delta) = A(\delta') = A(\delta'')$ and $p <_\alpha q, q <_\alpha r$. By transitivity of $<_\alpha$ we obtain $p <_\alpha r$ and $(\delta, p) <_P (\delta'', r)$ by (P1).

(P1)–(P2): we have $A(\delta) = A(\delta')$ and $A(\delta'') < A(\delta')$; so $A(\delta'') < A(\delta)$ and $(\delta, p) <_P (\delta'', r)$ by (P2).

(P1)–(P3), (P1)–(P4) and (P1)–(P5) are impossible as $\Delta_1 \cap \Delta_2 = \emptyset$ and thus $\delta' \in \Delta_1$ and $\delta' \in \Delta_2$ is impossible.

(P2)–(P1): $A(\delta') < A(\delta)$ and $A(\delta'') = A(\delta')$ gives $A(\delta'') < A(\delta)$ and so $(\delta, p) <_P (\delta'', r)$ by (P2).

(P3)–(P1): Here we have $\delta \in \Delta_1, \delta', \delta'' \in \Delta_2$. Therefore $(\delta, p) <_P (\delta'', r)$ by (P3).

(P4)–(P1), (P5)–(P1) and (P2)–(P4), (P2)–(P5) are impossible.

(P3)–(P2): $\delta \in \Delta_1, \delta', \delta'' \in \Delta_2$ and so $(\delta, p) <_P (\delta'', r)$ by (P3).

The combinations (P4)–(P2), (P5)–(P2), (P3)–(P3), (P3)–(P4) and (P3)–(P5) are all impossible.

(P4)–(P3): we have $\delta, \delta' \in \Delta_1$ and $\delta'' \in \Delta_2$ and therefore $(\delta, p) <_P (\delta'', r)$ by (P3).

(P5)–(P3): the same as for (P4)–(P3).

(P4)–(P4): $<_{\Delta_0}$ is transitive.

(P4)–(P5): We have $\delta <_{\Delta_0} \delta'$ and $\delta'' = \delta'$; so $(\delta, p) <_P (\delta'', r)$ by (P4).

(P5)–(P4): $\delta = \delta'$ and $\delta' <_{\Delta_0} \delta''$; therefore $(\delta, p) <_P (\delta'', r)$ by (P4).

(P5)–(P5): Here $\delta = \delta' = \delta''$, $p <_\alpha q, q <_\alpha r$. By transitivity of $<_\alpha$ we get $p <_\alpha r$ and $(\delta, p) <_P (\delta'', r)$ by (P5).

Now assume that $<_P$ is not well-founded. Then there exists an infinite sequence $\eta\colon (\delta_i, p_i)_{i \in \mathbb{N}}$ such that $(\delta_{i+1}, p_{i+1}) <_P (\delta_i, p_i)$ for all $i \in \mathbb{N}$. As Δ_0 is finite there exists a $\delta \in \Delta_0$ appearing infinitely often in η. That means there exists an infinite subsequence η^* of η which is of the form

$$(\delta, q_0), (\delta, q_1), \ldots (\delta, q_i), (\delta, q_{i+1}), \ldots$$

and $(\delta, q_{i+1}) <_P^* (\delta, q_i)$ for $i \in \mathbb{N}$ and the transitive closure $<_P^*$ of $<_P$. We have shown that $<_P$ is transitive and therefore $<_P = <_P^*$, hence $(\delta, q_{i+1}) <_P (\delta, q_i)$ for $i \in \mathbb{N}$.
It remains to distinguish two cases:

- $\delta \in \Delta_1$: then, for all $i \in \mathbb{N}$, $(\delta, q_{i+1}) <_P (\delta, q_i)$ by (P5) and (for $A(\delta) = \alpha$) $q_{i+1} <_\alpha q_i$ for all $i \in \mathbb{N}$. But this is impossible because $<_\alpha$ is well-founded.
- $\delta \in \Delta_2$: analogous to $\delta \in \Delta_1$ with (P1) instead of (P5). Again we obtain a contradiction to the well-foundedness of $<_\alpha$.

We have shown that η does not exist and $<_P$ is well-founded. □

Definition 40 *(Canonic point transition system)* Let $\mathbf{P}^\star\colon (\delta_0, \Delta^*, \Delta_e, \mathbf{P})$ be a point transition system. $\mathbf{P}^\star$ is called *canonic*

$$\text{if for all } (\delta, p) \to \mathcal{P}\colon C \in \mathbf{P} \text{ we have } (\delta', p') <_P (\delta, p) \text{ for all } (\delta', p') \in \mathcal{P}.$$

Theorem 1 (Termination) *Canonic point transition systems are terminating, i.e. if $\mathcal{P}^\star$ is a canonic point transition system and $\sigma \in \mathcal{S}$ then $T_{\mathcal{P}^\star}[\sigma]$ is finite.*

Proof By contradiction. Assume that $\mathbf{P}^\star$ is a canonic point transition system and $T_{\mathbf{P}^\star}[\sigma]$ is infinite for some $\sigma \in \mathcal{S}$. Then $T_{\mathbf{P}^\star}[\sigma]$ is an infinite tree with finite node degree. By König's lemma $T_{\mathbf{P}^\star}[\sigma]$ has an infinite path η for

$$\eta = (\delta_0, p_0), \ldots, (\delta_n, p_n), (\delta_{n+1}, p_{n+1}), \ldots$$

By definition of $T_{\mathcal{P}^\star}[\sigma]$, for each $(\delta_i, p_i), (\delta_{i+1}, p_{i+1})$ in η, there exists a point transition $(\delta_i, p) \to \mathcal{P} \colon C$ in $\mathcal{P}$ "matching" (δ_i, p_i). More precisely, there exists a parameter assignment σ' such that $\sigma'(p){\downarrow}= p_i$, $\sigma'[C] = \top$ and (δ_{i+1}, p_{i+1}) is obtained via a $(\delta_{i+1}, q) \in \mathcal{P}$ and $p_{i+1} = \sigma'(q)\downarrow$. As $\mathcal{P}^*$ is canonic we have $(\delta_{i+1}, q) <_P (\delta_i, p)$. By Proposition 13 we obtain $(\delta_{i+1}, \sigma'(q){\downarrow}) <_P (\delta_i, \sigma'(p){\downarrow})$ what yields just $(\delta_{i+1}, p_{i+1}) <_P (\delta_i, p_i)$ for all $i \in \mathbb{N}$. But by Proposition 14 $<_P$ is a well-ordering contradicting the existence of the infinite chain η. Therefore there exists no infinite σ-chain in $\mathcal{P}^*$ and $T_{\mathbf{P}^\star}[\sigma]$ is finite. □

Example 19 Let $\mathbf{P}^\star \colon (\delta_1, \Delta^*, \Delta_e, \mathbf{P})$ be the point transition system corresponding to Example 15 for $\Delta^* = \{\delta_1, \ldots, \delta_8\}$ and $\Delta_e = \{\delta_4, \ldots, \delta_8\}$. Here $\mathbf{P}$ consists of the following point transition rules:

$$
\begin{aligned}
&1\colon (\delta_1, (n, m)) \to \{(\delta_1, (n-1, m)), (\delta_2, (n, m-1))\} \colon m > 0 \wedge n > 0,\\
&2.\colon (\delta_1, (n, m)) \to \{(\delta_4, (n, m))\} \colon m > 0 \wedge n = 0,\\
&3\colon (\delta_1, (n, m)) \to \{(\delta_5, (n, m))\} \colon m = 0 \wedge n > 0,\\
&4\colon (\delta_1, (n, m)) \to \{(\delta_6, (n, m))\} \colon m = 0 \wedge n = 0,\\
&5\colon (\delta_2, (n, m)) \to \{(\delta_1, (n-1, m)), (\delta_3, (n, n-1, m))\} \colon n > 0,\\
&6\colon (\delta_2, (n, m)) \to \{(\delta_7, (n, m))\} \colon n = 0,\\
&7\colon (\delta_3, (k, n, m)) \to \{(\delta_3, (k-1, n, m))\} \colon k > 0,\\
&8\colon (\delta_3, (k, n, m)) \to \{(\delta_8, (k, n, m)\} \colon k = 0.
\end{aligned}
$$

It is easy to see that $\mathbf{P}^\star$ is canonic when we define $\Delta_1 = \{\delta_4, \ldots, \delta_8\}$ and $\Delta_2 = \{\delta_1, \delta_2, \delta_3\}$. Then, by definition of $<_P$, we have $\delta_i <_P \delta_j$ for $i \in \{4, \ldots, 8\}$, $j \in \{1, 2, 3\}$. Therefore the transitions 2,3,4,6 and 8 are decreasing via (P3). Transition 1 is decreasing via (P1), transition 5 via (P1) and (P2); finally transition 7 is decreasing via (P1). Therefore $T_{\mathbf{P}^\star}[\sigma]$ is finite for all $\sigma \in \mathcal{S}$ and, in particular, the function f in Example 15 is total and thus recursive. Also $T(\mathbf{P}, (\delta_2, (n, m)), \sigma)$ is finite for all $\sigma \in \mathcal{S}$, so the function g is recursive as well.

6 Schematic RPL$_0^\Psi$ Derivations

Schematic RPL$_0^\Psi$ derivations are extension of RPL$_0^\Psi$ derivations where, besides the formulas to be refuted, new kinds of axioms (in the form of labeled sequents) are included. These labeled sequents serve the purpose to establish recursive call structures in the proof. For constructing schematic RPL$_0^\Psi$ derivations we introduce a countably infinite set Δ of *proof symbols* which are used to label the individual proofs of a *proof schema*. A particular proof schema uses a finite set of *proof symbols* $\Delta^* \subset \Delta$. Like for point transition systems we assign an arity $A(\delta)$ to every $\delta \in \Delta^*$; $A(\delta)$ is just the arity of the input parameters for the proof labeled by δ. Also, we need a concept of *proof labels* which serve the purpose to relate some leafs of the proof tree to recursive calls.

Definition 41 *(Proof label)* Let $\delta \in \Delta$ and ϑ be a parameter substitution. Then the pair (δ, ϑ) is called a proof label.

We will need to locate specific occurrences of sequents in a proof π; to this aim we define Λ_π as the set of all sequent occurrences in π. By $|\pi|_\lambda$ we denote the sequent occurring at position λ for $\lambda \in \Lambda_\pi$. Furthermore, let ρ be a proof with the end-sequent $|\pi|_\lambda$; then by $\pi[\rho]_\lambda$ we denote the proof which results from replacing in π the derivation at λ by ρ.

Definition 42 *(Labeled sequents and derivations)* Let S be a sequent and (δ, ϑ) a proof label, them $(\delta, \vartheta)\colon S$ is a *labeled sequent*. A *labeled* RPL_0^{Ψ} derivation is an RPL_0^{Ψ} derivation π where all leaves are labeled. We distinguish *axiom labels* and *nonaxiom labels*. By $\mathrm{Axiom}(\pi)$ we denote the set of occurrences of leaves labeled by axiom labels, the set of the occurrences of leaves with nonaxiom labels is denoted by $\mathrm{Naxiom}(\pi)$. The set of all occurrences of leaves in π is denoted by $\mathrm{leaves}(\pi)$; so $\mathrm{leaves}(\pi) = \mathrm{Axiom}(\pi) \cup \mathrm{Naxiom}(\pi)$ and $\mathrm{Axiom}(\pi) \cap \mathrm{Naxiom}(\pi) = \emptyset$.

We will define proof schemata in the spirit of point transition systems. Before giving a formal definition the following example should serve the purpose to reveal the intuition behind the concept.

Example 20 We define a refutation schema for the theory

$$\Psi = (\{\hat{P}, \hat{Q}, \hat{f}\}, \hat{Q}, \{D(\hat{P}), D(\hat{Q}), D(\hat{f})\})$$

provided in Example 5. The defining equations for $\hat{P}$ and $\hat{Q}$ are:

$$\begin{aligned}
\hat{P}(X, \bar{0}) =& \neg P(X(\bar{0}), \hat{f}(a, 0)) \\
\hat{P}(X, s(n)) =& \hat{P}(X, n) \vee \neg P(X(s(n)), \hat{f}(a, s(n))) \\
\hat{Q}(X, Y, n, \bar{0}) =& P(\hat{f}(Y(\bar{0}), \bar{0}), Y(\bar{1})) \wedge \hat{P}(X, n) \\
\hat{Q}(X, Y, n, s(m)) =& P(\hat{f}(Y(\bar{0}), s(m)), Y(\bar{1})) \wedge \hat{P}(X, n)
\end{aligned}$$

For $\hat{f}$ we have $\hat{f}(X, 0) = X(0)$, $\hat{f}(X, m+1) = g(X(m+1), \hat{f}(X, m))$.
We create a proof symbol δ_0 which stands for the refutation of $\hat{Q}(X, Y, n, m)$. For this refutation we consider the partition $n = 0$ (base case) and $n > 0$ (step case). In order to refute $\hat{Q}(X, Y, n, m)$ for $n > 0$ we will need an auxiliary deduction which deduces $\hat{Q}(X, Y, 0, m)$ from the axioms $\hat{Q}(X, Y, n, m)$. In defining this derivation, which we give the label δ_1, we will need additional parameters k, l which do not occur in the definition of $\hat{Q}$ but are needed to control the recursions. For every proof symbol we have a parameter tuple, it is (n, m) for δ_0 and (n, m, k, l) for δ_1. For δ_0 we define the proof schema via the reduction system $\Pi(\delta_0, X, Y, n, m)$ and the specification of the end-sequent $S(\delta_0, X, Y, n, m) = \vdash$. We define

$$\begin{aligned}
\Pi(\delta_0, X, Y, n, m) = \{&(\delta_0, X, Y, n, m) \to \rho_0(\delta_0, X, Y, n, m)\colon n = 0, \\
&(\delta_0, X, Y, n, m) \to \rho_1(\delta_0, X, n, m)\colon n > 0\}.
\end{aligned}$$

which means that for $n = 0$ we select the proof $\rho_0(\delta_0, X, Y, n, m)$ for $n > 0$ the proof $\rho_1(\delta_0, X, n, m)$. By $\rho_0(\delta_0, X, Y, n, m)$ we denote the following derivation:

$$\dfrac{\dfrac{\dfrac{\vdash (\delta_1', (0, m))\colon \hat{Q}(X, Y, 0, m)}{\vdash P(\hat{f}(Y(0), m), Y(1)) \wedge \hat{P}(X, 0)}\,B\hat{Q}r}{\vdash P(\hat{f}(Y(0), m), Y(1))}\,\wedge_{r_1} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\vdash (\delta_1', (0, m))\colon \hat{Q}(X, Y, 0, m)}{\vdash P(\hat{f}(Y(0), m), Y(1)) \wedge \hat{P}(X, 0)}\,B\hat{Q}r}{\vdash \hat{P}(X, 0)}\,\wedge_{r_2}}{\vdash \neg P(X(0), \hat{f}(a, 0))}\,B\hat{P}r}{P(X(0), \hat{f}(a, 0)) \vdash}\,\neg\colon l}{\vdash}\,res(\sigma_1)$$

where $\sigma_1 = \{X(0) \leftarrow \hat{f}(Y(0), m), Y(1) \leftarrow \hat{f}(a, 0)\}$.Obviously, $\rho(\delta_0, X, Y, 0, m)$ is a RPL_0^{Ψ} refutation of $\hat{Q}(X, Y, 0, m)$. The label $(\delta_1', \emptyset)$ means that δ_1' is an axiom label for the case $(0, m)$. By $\rho_1(\delta_0, X, Y, n, m)$ we denote the following derivation:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}(\delta_1, (n, m, n, 0))\\ \vdash \hat{Q}(X, Y, 0, m)\end{array}}{\vdash P(\hat{f}(Y(0), m), Y(1)) \wedge \hat{P}(X, 0)}\,B\hat{Q}r}{\vdash P(\hat{f}(Y(0), m), Y(1))}\,\wedge_{r_1} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}(\delta_1, (n, m, n, 0))\\ \vdash \hat{Q}(X, Y, 0, m)\end{array}}{\vdash P(\hat{f}(Y(0), m), Y(1)) \wedge \hat{P}(X, 0)}\,B\hat{Q}r}{\vdash \hat{P}(X, 0)}\,\wedge_{r_2}}{\vdash \neg P(X(0), \hat{f}(a, 0))}\,B\hat{P}r}{P(X(0), \hat{f}(a, 0)) \vdash}\,\neg : l}{\vdash}\,res\sigma_1$$

In contrast to $\rho_0(\delta_0, X, Y, n, m)$ $\hat{Q}(X, Y, 0, m)$ is not an axiom in the proof but rather the end-sequent. The label $(\delta_1, (n, m, n, 0))$ represents a call of the proof $\rho(\delta_1, X, Y, m, n, k, l)$ where k is replaced by n and l by 0. For the proof symbol δ_1 we select the partition $\{n = 0,\ n > 0 \wedge l < n,\ n > 0 \wedge l \geq n\}$, define a reduction system $\Pi(\delta_1, X, Y, m, n, k, l)$ and a sequent $S(\delta_1, X, Y, n, m, k, l) = \hat{Q}(X, Y, l, m)$. We define $\Pi(\delta_1, X, Y, n, m, k, l) =$

$$\begin{aligned}
&\{(\delta_1, X, Y, n, m, k, l) \rightarrow \rho_0(\delta_1, X, Y, n, m, k, l)\colon n = 0,\\
&(\delta_1, X, Y, n, m, k, l) \rightarrow \rho_1(\delta_1, X, Y, n, m, k, l)\colon n > 0 \wedge l < n,\\
&(\delta_1, X, Y, n, m, k, l) \rightarrow \rho_2(\delta_1, X, Y, n, m, k, l)\colon n > 0 \wedge l \geq n\}.
\end{aligned}$$

where

$$\begin{aligned}
\rho_0(\delta_1, X, Y, n, m, k, l) &= (\delta_2', (n, m, k, l))\colon Q(X, Y, n, m),\\
\rho_2(\delta_1, X, Y, n, m, k, l) &= (\delta_3', (n, m, k, l))\colon Q(X, Y, n, m).
\end{aligned}$$

$(\delta_2', (n, m, k, l))$ and $(\delta_3', (n, m, k, l))$ are axiom labels where the computation of the proof stops. The most involved proof is $\rho_1(\delta_1, X, Y, m, n, k, l)$ which performs the real induction. It derives the end sequent $\vdash \hat{Q}(X, Y, l, m)$ from the sequent $(\delta_1, (m, n, p(k), s(l)))\colon \hat{Q}(X, Y, s(l), m)$. If $l < n$ then $(m, n, p(k), s(l))$ represents a further call of $\rho_1(\delta_1, X, Y, m, n, k, l)$. By this call the leaf $\hat{Q}(X, Y, s(l), m)$ is replaced by a derivation of $\hat{Q}(X, Y, s(l), m)$ with leaves $\hat{Q}(X, Y, ss(l), m)$ and so on. If the counter reaches $l = n$ the computation stops and we have reached the axioms $\hat{Q}(X, Y, n, m)$. Note that, in the whole proof system, we start with $l = 0$ and $k = n$ via the call from δ_0. That is, via the initiation given by the call from δ_0, we obtain a derivation of $\vdash \hat{Q}(X, Y, 0, m)$ from axioms $\vdash \hat{Q}(X, Y, n, m)$. Putting δ_0 and δ_1 together where δ_0 is the main symbol we eventually obtain a refutation of $\vdash \hat{Q}(X, Y, n, m)$.

By $\rho_1(\delta_1, X, Y, m, n, k, l)$ we denote the following derivation:

$$\dfrac{\dfrac{\begin{array}{c}(\delta_1, (n, m, p(k), s(l)))\\ \vdash \hat{Q}(X, Y, s(l), m)\end{array}}{\vdash P(\hat{f}(Y(0), m), Y(1)) \wedge \hat{P}(X, s(l))}\,S\hat{Q}r}{\vdash P(\hat{f}(Y(0), m), Y(1))}\,\wedge_{r_1} \tag{2}$$

$$\dfrac{\begin{matrix}(2)\\ \vdash P(\hat{f}(Y(0),m),Y(1))\end{matrix} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}(\delta_1,(n,m,p(k),s(l)))\\ \vdash \hat{Q}(X,Y,s(l),m)\end{matrix}}{\vdash P(\hat{f}(Y(0),m),Y(1)) \wedge \hat{P}(X,s(l))}\, S\hat{Q}r}{\vdash \hat{P}(X,s(l))}\, \wedge_{r_2}}{\vdash \neg P(X(l),\hat{f}(a,l)) \vee \hat{P}(X,l)}\, S\hat{P}r}{\vdash \neg P(X(l),\hat{f}(a,l)), \hat{P}(X,l)}\, \vee_r}{P(X(l),\hat{f}(a,l)) \vdash \hat{P}(X,l)}\, \neg_r}{\begin{matrix}\vdash \hat{P}(X,l)\\ (1)\end{matrix}}\, res(\sigma_2)$$

where $\sigma_2 = \{X(l) \leftarrow \hat{f}(Y(0),m), Y(1) \leftarrow \hat{f}(a,l)\}$.

$$\dfrac{\dfrac{\begin{matrix}(2)\\ \vdash P(\hat{f}(Y(0),m),Y(1))\end{matrix} \quad \begin{matrix}(1)\\ \vdash \hat{P}(X,l)\end{matrix}}{\vdash P(\hat{f}(Y(0),m),Y(1)) \wedge \hat{P}(X,l)}\, \wedge : Ir}{\vdash \hat{Q}(X,Y,l,m)}\, S\hat{Q}r^{+}$$

Definition 43 *(*RPL_0^{Ψ} *schema)* Let $\mathcal{D}$ be the tuple $(\Psi, \delta_0, \Delta^*, \Delta_a, \mathbf{X}, \Pi)$. $\mathcal{D}$ is called an RPL_0^{Ψ} *schema* if the following conditions are fulfilled:

- $\Psi : (\mathcal{S}, \hat{P}, \mathcal{T})$ is a theory as in Definition 3.
- Δ^* be a finite subset of Δ.
- $\delta_0 \in \Delta^*$, δ_0 is called the main symbol.
- $\Delta_a \subseteq \Delta^*$, the axiom symbols.
- $\mathbf{X}$ is a tuple of global variables.
- To every $\delta \in \Delta^*$ we assign a parameter tuple $\mathbf{n}_\delta$ of pairwise different parameters and a partition $\mathcal{C}_\delta = \{C_1^\delta, \ldots, C_{k(\delta)}^\delta\}$ of $\mathcal{S}$ where the C_i^δ contain only parameters in $\mathbf{n}_\delta$.
- Π is a set of pairs $\{(\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta), S(\delta, \mathbf{X}, \mathbf{n}_\delta)) \mid \delta \in \Delta^*\}$ where $S(\delta, \mathbf{X}, \mathbf{n}_\delta)$ is a sequent over the global variables $\mathbf{X}$ and the parameters in $\mathbf{n}_\delta$, and $\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta)$ is defined as follows:

  $$\{(\delta, \mathbf{X}, \mathbf{n}_\delta) \to \rho_1(\delta, \mathbf{X}, \mathbf{n}_\delta) : C_1^\delta, \ldots, (\delta, \mathbf{X}, \mathbf{n}_\delta) \to \rho_{k(\delta)}(\delta, \mathbf{X}, \mathbf{n}_\delta) : C_{k(\delta)}^\delta\}$$

  Note the following, for each $i \in \{1, \ldots, k(\delta)\}$ $\rho_i(\delta, \mathbf{X}, \mathbf{n}_\delta)$ is a labeled RPL_0^{Ψ} derivation of $S(\delta, \mathbf{X}, \mathbf{n}_\delta)$ and for each leaf $\lambda \in \mathrm{leaves}(\rho_i(\delta, \mathbf{X}, \mathbf{n}_\delta))$ we have

  - $|\rho_i(\delta, \mathbf{X}, \mathbf{n}_\delta)|_\lambda = (\delta', \mathbf{n}_\delta) : \vdash \hat{P}(\mathbf{Z}, \mathbf{r})$ for $\lambda \in \mathrm{Axiom}(\rho_i(\delta, \mathbf{X}, \mathbf{n}_\delta))$ where $\hat{P}(\mathbf{Z}, \mathbf{r})$ is a variant of the main atom of Ψ, $\delta' \in \Delta_a$, $A(\delta') = A(\delta)$ and $\mathbf{Z}$ is a tuple of global variables which is a subtuple of $\mathbf{X}$.
  - $|\rho_i(\delta, \mathbf{X}, \mathbf{n}_\delta)|_\lambda = (\delta', \mathbf{s}) : S(\delta', \mathbf{X}, \mathbf{s})$ for $\lambda \in \mathrm{Naxiom}(\rho_i(\delta, \mathbf{X}, \mathbf{n}_\delta))$ where $\delta' \in \Delta^* \backslash \Delta_a$, and $\mathbf{s}$ is an $A(\delta')$ tuple of terms in T_1^ω containing only parameters in $\mathbf{n}_\delta$.

Furthermore,

- $\mathbf{n}_\delta = (\mathbf{r}, \mathbf{m}_\delta)$, the first part being the parameter tuple $\mathbf{r}$ for the main symbol of the theory Ψ. The arity of $\mathbf{m}_\delta$ depends on δ.
- $\Pi(\delta_0, \mathbf{X}, \mathbf{n}_{\delta_0}) \neq \emptyset$,
- for all $\delta \in \Delta_a$ $\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta) = \emptyset$,

$\mathcal{D}$ is called a *refutation schema* of Ψ if $S(\delta_0, \mathbf{X}, \mathbf{n}_{\delta_0}) = \vdash$.

Remark 8 In defining $\mathbf{n} = (\mathbf{r}, \mathbf{m}_\delta)$ we distinguish the *static parameters* in $\mathbf{r}$ (which are used in the recursive formula definition) and the *dynamic parameters* in $\mathbf{m}_\delta$ which are used for carrying out the inductive steps. In Example 20 δ_0 has no dynamic parameters, but δ_1 has the dynamic parameters k and l. In a recursive call only dynamic parameters are substituted. The tuple $\mathbf{Z}$ of global variables is chosen in a way to guarantee essential disjointness of global variables in the resolution steps.

Given a parameter assignment σ, every proof schema defines a sequence of proofs in the following way: either we arrive at an axiom and stop or we find a leaf in the proof of the form $(\delta, \mathbf{p})\colon S(\delta, \mathbf{X}, \mathbf{p})$; in the latter case we identify the right condition C_i^δ and the proof $\rho_i(\delta, \mathbf{X}, \mathbf{p})$ and replace the leaf by the derivation $\rho_i(\delta, X, \mathbf{p})$. If this sequence converges we obtain an RPL_0^Ψ proof corresponding to σ. The formal definition is given below.

Definition 44 *(Semantics of proof schemata)* Let $\mathcal{D}$ be a proof schema

$$(\Psi, \delta_0, \Delta^*, \Delta_a, \mathbf{X}, \Pi).$$

Let $\sigma \in \mathcal{S}$; we define define a sequence $\mathcal{D}(\sigma, \alpha)_{\alpha \in \mathbb{N}}$ and call it the *proof sequence corresponding to* σ. Let

$$\Pi(\delta_0, \mathbf{X}, \mathbf{n}_{\delta_0}) = \{(\delta_0, \mathbf{X}, \mathbf{n}_{\delta_0}) \to \rho_1(\delta_0, \mathbf{X}, \mathbf{n}_{\delta_0})\colon C_1^{\delta_0},$$
$$\ldots, (\delta, \mathbf{X}, \mathbf{n}_{\delta_0}) \to \rho_{k(\delta_0)}(\delta_0, \mathbf{X}, \mathbf{n}_{\delta_0})\colon C_{k(\delta_0)}^{\delta_0}\}.$$

As $\{C_1^{\delta_0}, \ldots, C_{k(\delta_0)}^{\delta_0}\}$ is a partition there is exactly one $i \in \{1, \ldots, k(\delta_0)\}$ such that $\sigma[C_i^{\delta_0}] = \top$. Let $\sigma(\mathbf{n}_{\delta_0}) = \mathbf{k}$ where $\mathbf{k} = (\mathbf{r}_0, \mathbf{s})$ for $\sigma(\mathbf{r})\downarrow_\omega = \mathbf{r}_0$. Then we define

$$\mathcal{D}(\sigma, 0) = \rho_i(\delta_0, \mathbf{X}, \mathbf{k}).$$

Note that $\rho_i(\delta_0, \mathbf{X}, \mathbf{k})$ is an RPL_0^Ψ derivation of $S(\delta_0, X, \mathbf{k})$ and the leaves of $\rho_i(\delta_0, \mathbf{X}, \mathbf{k})$ are either axioms of the form $(\delta_0', \mathbf{k})\colon \hat{P}(\mathbf{X}, \mathbf{r}_0)$ for $\delta_0' \in \Delta_a$, or they are of the form $(\delta, \mathbf{l})\colon S(\delta, \mathbf{X}, \mathbf{l})$ where $\delta \in \Delta^* \backslash \Delta_a$ and $\mathbf{l}$ is a ground term tuple obtained in replacing the original one by substituting parameters by numerals.
Assume that $\mathcal{D}(\sigma, \alpha)$ is already defined. We distinguish two cases:

(a) All leaves in $\mathcal{D}(\sigma, \alpha)$ are of the form $(\delta, \mathbf{p})\colon \hat{P}(\mathbf{Z}, \mathbf{s})$ for $\delta \in \Delta_a$ where $\hat{P}$ is the main symbol of Ψ. Then we define $\mathcal{D}(\sigma, \alpha + 1) = \mathcal{D}(\sigma, \alpha)$.
(b) There are leaves of the form $(\delta, \mathbf{k})\colon S(\delta, \mathbf{X}, \mathbf{k})$ for an $A(\delta)$ ground term tuple $\mathbf{k}$ and $\delta \in \Delta^* \backslash \Delta_a$. For each leaf of this form we consider

$$\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta) = \{(\delta, \mathbf{X}, \mathbf{n}_\delta) \to \rho_1(\delta, \mathbf{X}, \mathbf{n}_\delta)\colon C_1^\delta,$$
$$\ldots, (\delta, \mathbf{X}, \mathbf{n}_\delta) \to \rho_{k(\delta)}(\delta, \mathbf{X}, \mathbf{n}_\delta)\colon C_{k(\delta)}^\delta\}.$$

Let $\sigma' \in \mathcal{S}$ such that $\sigma'(\mathbf{n}_\delta) = \mathbf{k}$. As $\{C_1^\delta, \ldots, C_{k(\delta)}^\delta\}$ is a partition there is exactly one $i \in \{1, \ldots, k(\delta)\}$ such that $\sigma'[C_i^\delta] = \top$. Now we have to select the production

$$(\delta, \mathbf{X}, \mathbf{n}_\delta) \to \rho_i(\delta, \mathbf{X}, \mathbf{n}_\delta)\colon C_i^\delta$$

from $\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta)$. Finally we replace the leaf $(\delta, \mathbf{k})\colon S(\delta, \mathbf{X}, \mathbf{k})$ by the RPL_0^Ψ derivation $\rho_i(\delta, \mathbf{X}, \mathbf{k})$. We perform this transformation for all leaves with $\delta \in \Delta^* \backslash \Delta_a$ and obtain a new RPL_0^Ψ derivation which we call $\mathcal{D}(\sigma, \alpha + 1)$.

If the sequence converges, i.e. $\mathcal{D}(\sigma, \alpha) = \mathcal{D}(\sigma, \alpha + 1)$ for some α we obtain an RPL_0^Ψ proof as a result. Of course it is desirable to obtain such a proof for all parameter assignments σ.

Definition 45 Let $\varphi = \mathcal{D}(\sigma,\alpha)$ where $\mathcal{D}(\sigma,\alpha) = \mathcal{D}(\sigma,\alpha+1)$. Then we say that $\mathcal{D}(\sigma,\alpha)_{\alpha\in\mathbb{N}}$ converges and converges to φ. $\mathcal{D}$ is called *convergent* if, for all $\sigma \in \mathcal{S}$, $\mathcal{D}(\sigma,\alpha)_{\alpha\in\mathbb{N}}$ converges.

If $\mathcal{D}(\sigma,\alpha)_{\alpha\in\mathbb{N}}$ converges then it converges to a proof $\varphi \in \mathrm{RPL}_0^{\Psi}$ which is parameter free. But φ can be further reduced to an RPL_0 derivation $\varphi\!\downarrow$ via the method defined in Sect. 4. We can now extend the semantics to RPL_0-proofs:

Definition 46 Let $\mathcal{D}$ be a proof schema which, for a $\sigma \in \mathcal{S}$, converges to an RPL_0^{Ψ} derivation $\varphi(\sigma)$. Then $\varphi(\sigma)\!\downarrow$ is called the RPL_0 proof defined by $\mathcal{D}$ under σ. If $\mathcal{D}$ converges for all σ then $\{\varphi(\sigma)\!\downarrow \mid \sigma \in \mathcal{S}\}$ is called the RPL_0 proof set defined by $\mathcal{D}$.

Example 21 Let $\mathcal{D}$ be the proof schema from Example 20. We compute $\mathcal{D}(\sigma,\alpha)$ for two different $\sigma \in \mathcal{S}$. Let $\sigma_0(n) = \bar{0}$, $\sigma_0(m) = \bar{2}$ and $\sigma_0(k) = \bar{1}$ for all parameters $k \notin \{n,m\}$. We start with $\mathcal{D}(\sigma_0,0)$. As $\sigma_0(n) = \bar{0}$ we have to replace $(\delta_0, X, Y, (n,m))$ by the proof $\rho_0(\delta_0, X, Y, (\bar{0},\bar{2}))$. So we define $\mathcal{D}(\sigma_0,0) =$

$$\dfrac{\dfrac{\dfrac{\vdash (\delta_1',(\bar0,\bar2))\colon \hat{Q}(X,Y,\bar0,\bar2)}{\vdash P(\hat f(Y(\bar0),\bar2),Y(\bar1)) \wedge \hat P(X,\bar0)}\,B\hat{Q}r}{\vdash P(\hat f(Y(\bar0),\bar2),Y(\bar1))}\,\wedge_{r_1} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\vdash (\delta_1',(\bar0,\bar2))\colon \hat{Q}(X,Y,\bar0,\bar2)}{\vdash P(\hat f(Y(\bar0),\bar2),Y(\bar1)) \wedge \hat P(X,\bar0)}\,B\hat{Q}r}{\vdash \hat P(X,\bar0)}\,\wedge_{r_2}}{\vdash \neg P(X(\bar0),\hat f(a,\bar0))}\,B\hat{P}r}{P(X(\bar0),\hat f(a,\bar0)) \vdash}\,\neg:l}{\vdash}\,res(\sigma_1')$$

for $\sigma_1' = \{X(0) \leftarrow \hat f(Y(0),2), Y(1) \leftarrow \hat f(a,0)\}$. In $\rho_0(\delta_0, X, Y, (\bar0,\bar2))$ all leaves are labeled by the axiom labels δ_1' and there is no leaf to expand; thus $\mathcal{D}(\sigma_0,1) = \mathcal{D}(\sigma_0,0)$ and $\mathcal{D}(\sigma_0,\alpha)_{\alpha\in\mathbb{N}}$ converges to $\rho_0(\delta_0, X, Y, (0,2))$.
Now let $\sigma_1(p) = \sigma_0(p)$ for $p \neq n$ and $\sigma_1(n) = 1$. In this case the condition $n > 0$ holds and we have to choose the proof $\rho_1(\delta_0, X, Y, 1, 2)$. We obtain $\mathcal{D}(\sigma_1,0) =$

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}(\delta_1,(1,2,1,0))\\ \vdash \hat{Q}(X,Y,0,2)\end{array}}{\vdash P(\hat f(Y(0),m),Y(1)) \wedge \hat P(X,0)}\,B\hat{Q}r}{\vdash P(\hat f(Y(0),2),Y(1))}\,\wedge_{r_1} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}(\delta_1,(1,2,1,0))\\ \vdash \hat{Q}(X,Y,0,2)\end{array}}{\vdash P(\hat f(Y(0),2),Y(1)) \wedge \hat P(X,0)}\,B\hat{Q}r}{\vdash \hat P(X,0)}\,\wedge_{r_2}}{\vdash \neg P(X(0),\hat f(a,0))}\,B\hat{P}r}{P(X(0),\hat f(a,0)) \vdash}\,\neg:l}{\vdash}\,res(\sigma_1')$$

Now both leaves in $\rho_1(\delta_0, X, Y, 1, 2)$ represent a call to the proof labeled by δ_1. In $\Pi(\delta_1, X, Y, n, m, k, l)$ we have to choose the proof $\rho_1(\delta_1, X, Y, 1, 2, 1, 0)$ via $\sigma'(n) = 1$, $\sigma'(m) = 2$, $\sigma'(k) = 1$, $\sigma'(l) = 0$ (the second condition applies). So we obtain $\mathcal{D}(\sigma_1,1) =$

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\rho_1(\delta_1,X,Y,(1,2,1,0))\\ \vdash \hat{Q}(X,Y,0,2)\end{array}}{\vdash P(\hat f(Y(0),m),Y(1)) \wedge \hat P(X,0)}\,B\hat{Q}r}{\vdash P(\hat f(Y(0),2),Y(1))}\,\wedge_{r_1} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\rho_1(\delta_1,X,Y,(1,2,1,0))\\ \vdash \hat{Q}(X,Y,0,2)\end{array}}{\vdash P(\hat f(Y(0),2),Y(1)) \wedge \hat P(X,0)}\,B\hat{Q}r}{\vdash \hat P(X,0)}\,\wedge_{r_2}}{\vdash \neg P(X(0),\hat f(a,0))}\,B\hat{P}r}{P(X(0),\hat f(a,0)) \vdash}\,\neg:l}{\vdash}\,res\sigma_1$$

Now $\rho_1(\delta_1, X, Y, (1, 2, 1, 0)) =$

$$\begin{array}{c}\dfrac{\dfrac{\begin{array}{c}(\delta_1, (1,2,0,1))\\ \vdash \hat{Q}(X, Y, 1, 2)\end{array}}{\vdash P(\hat{f}(Y(0), 2), Y(1)) \wedge \hat{P}(X, 1)}\, S\hat{Q}r}{\vdash P(\hat{f}(Y(0), 2), Y(1))}\, \wedge_{r_1}\\ (2)\end{array}$$

$$\begin{array}{c}\dfrac{\begin{array}{c}(2)\\ \vdash P(\hat{f}(Y(0), 2), Y(1))\end{array} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}(\delta_1, (1,2,0,1))\\ \vdash \hat{Q}(X, Y, 1, 2)\end{array}}{\vdash P(\hat{f}(Y(0), 2), Y(1)) \wedge \hat{P}(X, 1)}\, S\hat{Q}r}{\vdash \hat{P}(X, 1)}\, \wedge_{r_2}}{\vdash \neg P(X(0), \hat{f}(a, 0)) \vee \hat{P}(X, 0)}\, S\hat{P}r}{\vdash \neg P(X(0), \hat{f}(a, 0)), \hat{P}(X, 0)}\, \vee_r}{P(X(0), \hat{f}(a, 0)) \vdash \hat{P}(X, 0)}\, \neg_r}{\vdash \hat{P}(X, 0)}\, res(\sigma_2)\\ (1)\end{array}$$

where $\sigma_2 = \{X(0) \leftarrow \hat{f}(Y(0), 2), Y(1) \leftarrow \hat{f}(a, 0)\}$.

$$\dfrac{\dfrac{\begin{array}{c}(2)\\ \vdash P(\hat{f}(Y(0), 2), Y(1))\end{array} \quad \begin{array}{c}(1)\\ \vdash \hat{P}(X, 0)\end{array}}{\vdash P(\hat{f}(Y(0), 2), Y(1)) \wedge \hat{P}(X, 0)}\, \wedge : Ir}{\vdash \hat{Q}(X, Y, 0, 2)}\, S\hat{Q}r^+$$

In $\mathcal{D}(\sigma, 1)$ we have the leaves $\vdash (\delta_1, (1, 2, 0, 1))\colon \hat{Q}(X, Y, 1, 2)$. Again we have to call δ_1, this time via σ'' for $\sigma''(n) = 1, \sigma''(m) = 2, \sigma''(k) = 0, \sigma''(l) = 1$. As now $\sigma''(l) = \sigma''(n)$ we have to call $\rho_2(\delta_1, X, Y, (1, 2, 0, 1))$. But

$$\rho_2(\delta_1, X, Y, 1, 2, 0, 1) = (\delta_3', (1, 2, 0, 1))\colon Q(X, Y, 1, 2)$$

where $\delta_3' \in \Delta_a$. So we obtain $\mathcal{D}(\sigma_1, 2)$ from $\mathcal{D}(\sigma_1, 1)$ by replacing the leaves $\vdash (\delta_1, (1, 2, 0, 1))\colon \hat{Q}(X, Y, 1, 2)$ by the axiom leaves $(\delta_3', (1, 2, 0, 1))\colon Q(X, Y, 1, 2)$. In $\mathcal{D}(\sigma_1, 2)$ all leaves are axiom leaves and so $\mathcal{D}(\sigma_1, 3) = \mathcal{D}(\sigma_1, 2)$. Therefore also $\mathcal{D}(\sigma_1, \alpha)_{\alpha \in \mathbb{N}}$ converges. It is easy to see that $\mathcal{D}$ itself is convergent. The means to prove this formally will be developed below.

The concept of proof schema $\mathcal{D}$ is defined in a way that the skeleton of $\mathcal{D}$ is, in fact, a point transition system: we just strip the schema of its logical part and obtain the remaining "call part" of the schema.

Definition 47 Let $\mathcal{D}\colon (\Psi, \delta_0, \Delta^*, \Delta_a, \mathbf{X}, \Pi)$ be a proof schema. Assume that for $\delta \in \Delta^*$ we have

$$\begin{aligned}\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta) = \{&(\delta, \mathbf{X}, \mathbf{n}_\delta) \rightarrow \rho_1(\delta, \mathbf{X}, \mathbf{n}_\delta)\colon C_1^\delta,\\ &\ldots, (\delta, \mathbf{X}, \mathbf{n}_\delta) \rightarrow \rho_{k(\delta)}(\delta, \mathbf{X}, \mathbf{n}_\delta)\colon C_{k(\delta)}^\delta\}.\end{aligned}$$

Let $t\colon (\delta, \mathbf{X}, \mathbf{n}_\delta) \rightarrow \rho_i(\delta, \mathbf{X}, \mathbf{n}_\delta)\colon C_i^\delta$ be a production in $\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta)$. Let $\Lambda(\delta, i)$ the set of all leaves in the proof $\rho_i(\delta, \mathbf{X}, \mathbf{n}_\delta)$. For every $\lambda \in \Lambda(\delta, i)$ of the form $\lambda\colon (\delta', \mathbf{p})\colon F$ we define $pts(\lambda) = (\delta', \mathbf{p})$.

Then we replace the production t in $\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta)$ by

$$pts(t) = (\delta, \mathbf{n}_\delta) \rightarrow \{pts(\lambda) \mid \lambda \in \Lambda(\delta, i)\}$$

and $\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta)$ by

$$pts(\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta)) = \{pts(t) \mid t \in \Pi(\delta, \mathbf{X}, \mathbf{n}_\delta)\}.$$

Finally we define $\mathbf{P}(\delta) = pts(\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta))$ for all $\delta \in \Delta^*$ and set

$$pts(\mathcal{D}) = (\delta_0, \Delta^*, \Delta_a, \mathbf{P}).$$

Proposition 15 *Let $\mathcal{D}\colon (\Psi, \delta_0, \Delta^*, \Delta_a, \mathbf{X}, \Pi)$ be a proof schema. Then $pts(\mathcal{D})$ is a point transition system.*

Proof Immediate by the Definitions 36, 43 and 47. □

Example 22 Let $\mathcal{D}$ be the proof schema in Example 20. Then $pts(\mathcal{D})$ is the point transition system $(\delta_0, \Delta^*, \Delta_a, \mathbf{P})$, where $\Delta^* = \{\delta_0, \delta_1, \delta'_1, \delta'_2, \delta'_3\}$, $\Delta_a = \{\delta'_1, \delta'_2, \delta'_3\}$ and $\mathbf{P}$ defined below:

$$\begin{aligned}
\mathbf{P}(\delta_0) = \{&(\delta_0, (n, m)) \rightarrow \{(\delta'_1, (0, m))\}\colon n = 0,\\
&(\delta_0, (n, m)) \rightarrow \{(\delta_1, (n, m, n, 0))\}\colon n > 0\}\\
\mathbf{P}(\delta_1) = \{&(\delta_1, (n, m, k, l)) \rightarrow \{(\delta'_2, (n, m, k, l))\}\colon n = 0,\\
&(\delta_1, (n, m, k, l)) \rightarrow \{(\delta_1, (n, m, p(k), s(l)))\}\colon n > 0 \wedge l < n,\\
&(\delta_1, (n, m, k, l)) \rightarrow \{(\delta'_3, (n, m, k, l))\}\colon n > 0 \wedge l \geq n\},
\end{aligned}$$

$$\mathbf{P}(\delta'_1) = \mathbf{P}(\delta'_2) = \mathbf{P}(\delta'_3) = \emptyset.$$

Note that the righthandsides of the productions in $pts(\mathcal{D})$ are sets—not multisets. Therefore we count different leaves with the same labels just once. It is easy to see that $pts(\mathcal{D})$ is a canonic point transition system and thus terminating. We will prove below that this implies that $\mathcal{D}$ is convergent.

Lemma 1 *Let $\mathcal{D}(\sigma, \alpha)$ be as in Definition 44 and $T(\mathbf{P}^\star, (\delta, p), \sigma, \alpha)$ as in Definition 37. Then, for any $\alpha \in \mathbb{N}$:*

(a) *if $(\delta, \mathbf{k})\colon S$ is a leaf node in $\mathcal{D}(\sigma, \alpha)$ then $(\delta, \mathbf{k})$ is a leaf node in $T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, \alpha + 1)$;*

(b) *if $(\delta, \mathbf{k})$ is a leaf node in $T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, \alpha + 1)$ then there exists a sequent S such that $(\delta, \mathbf{k})\colon S$ is a leaf node in $\mathcal{D}(\sigma, \alpha)$.*

Proof We prove (a) by induction on α. Let $\alpha = 0$. Let us consider

$$\begin{aligned}
\Pi(\delta_0, \mathbf{X}, \mathbf{n}_{\delta_0}) = \{&(\delta_0, \mathbf{X}, \mathbf{n}_{\delta_0}) \rightarrow \rho_1(\delta_0, \mathbf{X}, \mathbf{n}_{\delta_0})\colon C_1^{\delta_0},\\
&\ldots, (\delta, \mathbf{X}, \mathbf{n}_{\delta_0}) \rightarrow \rho_{k(\delta_0)}(\delta_0, \mathbf{X}, \mathbf{n}_{\delta_0})\colon C_{k(\delta_0)}^{\delta_0}\},
\end{aligned}$$

and $\sigma[C_i^{\delta_0}] = \top$. Then $\mathcal{D}(\sigma, 0)$ is defined as $\rho_i(\delta_0, \mathbf{X}, \sigma(\mathbf{n}_{\delta_0}))$. Now let $(\delta_j, \mathbf{p}_j)\colon F_j$ be the leaves in in $\rho_i(\delta_0, \mathbf{X}, \mathbf{n}_{\delta_0})$ for $j = 1, \ldots, \beta$; then $(\delta_j, \sigma(\mathbf{p}_j))\colon \sigma(F_j))$ are the leaves in $\rho_i(\delta_0, \mathbf{X}, \sigma(\mathbf{n}_{\delta_0}))$. By Definition 47 $pts(\Pi(\delta_0, \mathbf{X}, \mathbf{n}_{\delta_0}))$ contains the production

$$(\delta_0, \mathbf{n}_{\delta_0}) \rightarrow \{(\delta_1, \mathbf{p}_1), \ldots, (\delta_\beta, \mathbf{p}_\beta)\}.$$

Hence, by Definition 37 $T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, 1)$ has exactly the leaves $(\delta_j, \sigma(\mathbf{p}_j))$. This concludes the case $\alpha = 0$.
Now assume that, for any leaf $(\delta, \mathbf{k})\colon F$ in $\mathcal{D}(\sigma, \alpha)$ there exists a leaf $(\delta, \mathbf{k})$ in $T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, \alpha+1)$. If $\delta \in \Delta_a$ then, by definition, $(\delta, \mathbf{k})\colon F$ is a leaf in $\mathcal{D}(\sigma, \alpha+1)$ and $(\delta, \mathbf{k})$ is a leaf in $T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, \alpha+2)$. Otherwise assume that $\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta) \neq \emptyset$ and

$$\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta) = \{(\delta, \mathbf{X}, \mathbf{n}_\delta) \to \rho_1(\delta, \mathbf{X}, \mathbf{n}_\delta)\colon C_1^\delta, \\ \ldots, (\delta, \mathbf{X}, \mathbf{n}_\delta) \to \rho_{k(\delta)}(\delta, \mathbf{X}, \mathbf{n}_\delta)\colon C_{k(\delta)}^\delta\}.$$

Let $\sigma' \in \mathcal{S}$ such that $\sigma'(\mathbf{n}_\delta) = \mathbf{k}$ and $\sigma'[C_i^\delta] = \top$. Now we have to select the production

$$(\delta, \mathbf{X}, \mathbf{n}_\delta) \to \rho_i(\delta, \mathbf{X}, \mathbf{n}_\delta)\colon C_i^\delta$$

from $\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta)$. Again, assume that $(\delta_j, \mathbf{p}_j)\colon F_j$ are the leaves in $\rho_i(\delta, \mathbf{X}, \mathbf{n}_\delta)$ for $j = 1, \ldots, \gamma$, and so $(\delta_j, \sigma'(\mathbf{p}_j))\colon \sigma'(F_j)$ are the leaves in in $\rho_i(\delta, \mathbf{X}, \sigma'(\mathbf{n}_\delta))$. By Definition 47 $pts(\Pi(\delta, \mathbf{X}, \mathbf{n}_\delta))$ contains the production

$$(\delta, \mathbf{n}_\delta) \to \{(\delta_1, \mathbf{p}_1), \ldots, (\delta_\beta, \mathbf{p}_\gamma)\}.$$

By Definition 37 , $T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, \alpha + 2)$ contains (new) leaves of the form $(\delta_j, \sigma'(\mathbf{p}_j))$.
The proof of (b) is analogous to that of (a). □

Theorem 2 *Let $\mathcal{D}$ be a proof schema. $\mathcal{D}$ is convergent iff $pts(\mathcal{D})$ is terminating.*

Proof $\mathcal{D}$ is convergent $\Rightarrow$:
For all $\sigma \in \mathcal{S}$ there exists an α such that $\mathcal{D}(\sigma, \alpha) = \mathcal{D}(\sigma, \alpha + 1)$. By definition of $\mathcal{D}$ this happens only if for all leaves $(\delta, \mathbf{k})\colon S$ in $\mathcal{D}(\sigma, \alpha)$ we have $\delta \in \Delta_a$. By Lemma 1 all leaves in $T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, \alpha+1)$ are of the form $(\delta, \mathbf{k})$ for $\delta \in \Delta_a$. As Δ_a is the set of all end labels in $pts(\mathcal{D})$ we get

$$T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, \alpha + 1) = T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, \alpha + 2)$$

and, for all σ, $pts(\mathcal{D})$ terminates on σ; thus $pts(\mathcal{D})$ is terminating.
$pts(\mathcal{D})$ is terminating $\Rightarrow$:
For all $\sigma \in \mathcal{S}$ there exists an α such that

$$T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, \alpha) = T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, \alpha + 1).$$

For all $\alpha > 0$ this implies that, for all leaves $(\delta, \mathbf{k})$ in $T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, \alpha)$, $\delta \in \Delta_a$. Now let $(\delta', \mathbf{p})\colon S$ be a leaf in $\mathcal{D}(\sigma, \alpha-1)$; by Lemma 1 $(\delta', \mathbf{p})$ is a leaf in $T(pts(\mathcal{D}), (\delta_0, \mathbf{n}_{\delta_0}), \sigma, \alpha)$ and, by assumption, $\delta' \in \Delta_a$. Therefore, by definition of $\mathcal{D}(\sigma, \alpha+1)$, we get $\mathcal{D}(\sigma, \alpha+1) = \mathcal{D}(\sigma, \alpha)$ and $\mathcal{D}$ converges on σ. □

We can now characterize the convergence of proof schemata via point transition systems:

Theorem 3 *Let $\mathcal{D}$ be a proof schema. Then $\mathcal{D}$ is convergent if $pts(\mathcal{D})$ is canonic.*

Proof Immediate by the Theorems 1 and 2. □

Now it is easily verified that the proof schema $\mathcal{D}$ from Example 20 is convergent. The point transition system $pts(\mathcal{D})$ shown in Example 22 is canonic: for the ordering $<_P$ we use the lexicographic tuple ordering and we partition Δ^* into Δ_1, Δ_2 by $\Delta_1 = \Delta_a$, $\Delta_2 = \{\delta_0, \delta_1\}$.

Example 23 Below is the complete refutation schema for the schematic formula provided in Example 6. In addition to the construction provided in Example 6 we require an additional global variable Y with a two arguments. Let $\mathcal{D} = (\Psi, \delta_0, \Delta^*, \Delta_a, \mathbf{X} \cup \{Y\}, \Pi)$ be an RPL_0^{Ψ} schema where Ψ is the theory presented in Example 6, $\Delta^*=\{\delta_0, \cdots, \delta_6, \delta_0^1, \delta_0^2, \delta_0^3, \delta_0^4, \delta_4', \delta_5'\}$, $\Delta_a = \{\delta_0^1, \delta_0^2, \delta_0^3, \delta_0^4, \delta_4', \delta_5'\}$, and

$$\Pi = \left\{ \begin{array}{c} (\Pi_0(\delta_0, \mathbf{X}, Y, n, m), \vdash) \\ (\Pi_1(\delta_1, \mathbf{X}, Y, n, m, w, k, r, q), \vdash f(Y(w,k)) < k) \\ \Big(\Pi_2(\delta_2, \mathbf{X}, Y, n, m, w, k, r, q), \vdash f(Y(w,k)) < k, \hat{F}_5(\mathbf{X}, k, q)\Big) \\ \Big(\Pi_3(\delta_3, \mathbf{X}, Y, n, m, w, k, r, q), \vdash \hat{F}_4(\mathbf{X}, k, q)\Big) \\ \Big(\Pi_4(\delta_4, \mathbf{X}, Y, n, m, w, k, r, q), \vdash\vdash \hat{F}_2(\mathbf{X}, k, q)\Big) \\ \Big(\Pi_5(\delta_5, \mathbf{X}, Y, n, m, w, k, r, q), \vdash \hat{F}_3(\mathbf{X}, k, q)\Big) \\ (\Pi_6(\delta_6, \mathbf{X}, Y, n, m, w, k, r, q), \vdash f(Y(w,k)) < k)) \end{array} \right\},$$

where,

- $\Pi_0(\delta_0, \mathbf{X}, Y, n, m) =$

$$\left\{ \begin{array}{l} (\delta_0, \mathbf{X}, Y, n, m) \rightarrow \rho_1(\delta_0, \mathbf{X}, Y, n, m) \colon n > 0 \wedge m > 0 \\ (\delta_0, \mathbf{X}, Y, n, m) \rightarrow \rho_2(\delta_0, \mathbf{X}, Y, n, m) \colon n = 0 \wedge m > 0 \\ (\delta_0, \mathbf{X}, Y, n, m) \rightarrow \rho_3(\delta_0, \mathbf{X}, Y, n, m) \colon n > 0 \wedge m = 0 \\ (\delta_0, \mathbf{X}, Y, n, m) \rightarrow \rho_4(\delta_0, \mathbf{X}, Y, n, m) \colon n = 0 \wedge m = 0 \end{array} \right\}.$$

  $\rho_4(\delta_0, \mathbf{X}, Y, n, m)$ is defined in Example 14.
  The other proofs are as follows:

  - $\rho_1(\delta_0, \mathbf{X}, Y, n, m)$

$$\dfrac{\begin{array}{c}(\delta_1, (n, m, n, n, s(m), 0)) \\ \vdash f(Y(n, 0)) < 0\end{array} \qquad \dfrac{\dfrac{\dfrac{\begin{array}{c}(\delta_5, (n, m, n, n, 0, m)) \\ \vdash \hat{F}_3(\mathbf{X}, 0, m)\end{array}}{\vdash \hat{F}_5(\mathbf{X}, 0, m) \wedge f(a) \not< 0}\, B\hat{F}_3 r}{\vdash f(a) \not< 0}\, \wedge : r_2}{f(a) < 0 \vdash}\, \neg : r}{\vdash}\, \mathrm{Res}(\mu)$$

  where $\mu = \{Y(n, 0) \leftarrow a\}$.
  - $\rho_2(\delta_0, \mathbf{X}, Y, n, m)$

$$\dfrac{\begin{array}{c}(\delta_1, (0, m, 0, 0, s(m), 0)) \\ \vdash f(Y(0, 0)) < 0\end{array} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\left(\delta_0^4, (0, m)\right) \\ \vdash \hat{F}_1(\mathbf{X}, 0, m)\end{array}}{\vdash \hat{F}_2(\mathbf{X}, 0, m) \wedge \hat{F}_3(\mathbf{X}, 0, m)}\, S\hat{F}_1 r}{\vdash \hat{F}_3(\mathbf{X}, 0, m)}\, \wedge : r}{\vdash \hat{F}_5(, \mathbf{X}, Y(0), 0, m) \wedge f(a) \not< 0}\, B\hat{F}_3 r}{\vdash f(a) \not< 0}\, \wedge : r_2}{f(a) < 0 \vdash}\, \neg : r}{\vdash}\, \mathrm{Res}(\mu)$$

  where $\mu = \{Y(0, 0) \leftarrow a\}$.
  - $\rho_3(\delta_0, \mathbf{X}, Y, n, m)$

$$\dfrac{\begin{matrix}(\delta_6,(n,0,n,n,s(0),0))\\ \vdash f(Y(n,0))<0\end{matrix} \qquad \dfrac{\dfrac{\dfrac{\begin{matrix}(\delta_5,(n,0,n,n,0,0))\\ \vdash \hat{F}_3(\mathbf{X},0,0)\end{matrix}}{\vdash \hat{F}_5(\mathbf{X},0,0)\wedge f(a)\not< 0}\,B\hat{F}_3r}{\vdash f(a)\not< 0}\,\wedge : r_2}{f(a)<0\vdash}\,\neg : r}{\vdash}\,\mathrm{Res}(\mu)$$

where $\mu=\{Y(n,0)\leftarrow a\}$.

- $\Pi_1(\delta_1,\mathbf{X},Y,n,m,w,k,r,q)=$

$$\left\{\begin{matrix}(\delta_1,\mathbf{X},Y,n,m,w,k,r,q)\rightarrow\rho_1(\delta_1,\mathbf{X},Y,n,m,w,k,r,q)\colon w>0\\ (\delta_1,\mathbf{X},Y,n,m,w,k,r,q)\rightarrow\rho_2(\delta_1,\mathbf{X},Y,n,m,w,k,r,q)\colon w=0\end{matrix}\right\}.$$

- $\rho_1(\delta_1,\mathbf{X},Y,n,m,w,k,r,q)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}(\delta_3,(n,m,w,k,p(r),0))\\ \vdash \hat{F}_4(\mathbf{X},k,0)\end{matrix}}{\vdash f((X_2(k,0))\not< s(k)\vee f(X_2(k,0))<k\vee f(X_2(k,0))\sim k}\,B\hat{F}_4r}{\vdash f(X_2(k,0))\not< s(k),\, f(X_2(k,0))<k\vee f(X_2(k,0))\sim k}\,\vee : r}{\vdash f(X_2(k,0))\not< s(k),\, f(X_2(k,0))<k,\, f(X_2(k,0))\sim k}\,\vee : r}{f(X_2(k,0))<s(k)\vdash f(X_2(k,0))<k,\, f(X_2(k,0))\sim k}\,\neg : r$$

$$(2)$$

$$\dfrac{\begin{matrix}(\delta_1,(n,m,p(w),s(k),r,0))\\ \vdash f(Y(p(w),s(k)))<s(k)\end{matrix}\qquad (2)}{\vdash f(Y(p(w),s(k)))<k,\, f(Y(p(w),s(k)))\sim k}\,\mathrm{Res}(\mu_1)$$

$$(1)$$

$$\dfrac{(1)\qquad \dfrac{\dfrac{\dfrac{\begin{matrix}(\delta_2,(n,m,w,k,p(p(r)),0))\\ \vdash f(Y(w,k))<k,\,\hat{F}_5(\mathbf{X},k,0)\end{matrix}}{\vdash f(Y(w,k))<k,\, f(\hat{S}(X_3(k),0))\nsim k}\,B\hat{F}_5r}{\vdash f(Y(w,k))<k,\, f(X_3(k))\nsim k}\,B\hat{S}r}{f(X_3(k))\sim k\vdash f(Y(w,k))<k}\,\neg : r}{\vdash f(Y(w,k))<k}\,\mathrm{Res}(\mu_2)$$

where $\mu_1=\left\{X_2(k,0)\leftarrow Y(p(w),s(k))\right\}$ and

$$\mu_2=\left\{Y(p(w),s(k))\leftarrow Y(w,k),\;X_3(k)\leftarrow Y(w,k)\right\}$$

.

- $\rho_2(\delta_1,\mathbf{X},Y,n,m,w,k,r,q)$

$$\dfrac{\dfrac{\dfrac{\begin{matrix}(\delta_4,(n,m,w,k,p(r),0))\\ \vdash \hat{F}_2(\mathbf{X},k,0)\end{matrix}}{\vdash f(X_1(k,0))<k\vee f(X_1(k,0))\sim k}\,B\hat{F}_2r}{\vdash f(X_1(k,0))<k,\, f(X_1(k,0))\sim k}\,\vee : r}{\vdash f(X_1(k,0))<k,\, f(X_1(k,0))\sim k}\,B\hat{S}r$$

$$(1)$$

$$\dfrac{(1) \qquad \dfrac{\dfrac{\dfrac{\begin{array}{c}(\delta_2, (n, m, 0, k, p(p(r)), 0))\\ \vdash f(Y(0, k)) < k,\, \hat{F}_5(\mathbf{X}, k, 0)\end{array}}{\vdash f(Y(0, k)) < k,\, f(\hat{S}(X_3(k), 0)) \nsim k}\; B\hat{F}_5 r}{\vdash f(Y(0, k)) < k,\, f(X_3(k)) \nsim k}\; B\hat{S} r}{f(X_3(k)) \sim k \vdash f(Y(0, k)) < k}\; \neg : r}{\vdash f(Y(0, k)) < k}\; \mathrm{Res}(\mu)$$

where $\mu = \left\{ X_1(k, 0) \leftarrow Y(0, k)\ ,\ X_3(k) \leftarrow Y(0, k) \right\}$.

- $\Pi_2(\delta_2, \mathbf{X}, Y, n, m, w, k, r, q) =$

$$\left\{ \begin{array}{l} (\delta_2, \mathbf{X}, Y, n, m, w, k, r, q) \rightarrow \rho_1(\delta_2, \mathbf{X}, Y, n, m, w, k, r, q) \colon r > 0 \\ (\delta_2, \mathbf{X}, Y, n, m, w, k, r, q) \rightarrow \rho_2(\delta_2, \mathbf{X}, Y, n, m, w, k, r, q) \colon r = 0 \end{array} \right\}.$$

- $\rho_1(\delta_2, \mathbf{X}, Y, n, m, w, k, r, q)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\big(\delta_3, (n, m, w, k, p(r), s(q))\big)\\ \vdash \hat{F}_4(\mathbf{X}, k, s(q))\end{array}}{\vdash (f(\hat{S}(X_2(k, s(q)), s(q))) \nless s(k) \vee f(X_2(k, s(q))) < k \vee f(\hat{S}(X_2(k, s(q)), s(q))) \sim k) \wedge \hat{F}_4(\mathbf{X}, k, q)}\; S\hat{F}_4 r}{\vdash f(\hat{S}(X_2(k, s(q)), s(q))) \nless s(k) \vee f(X_2(k, s(q))) < k \vee f(\hat{S}(X_2(k, s(q)), s(q))) \sim k}\; \wedge : r_2}{\vdash f(\hat{S}(X_2(k, s(q)), s(q))) \nless s(k),\, f(X_2(k, s(q))) < k \vee f(\hat{S}(X_2(k, s(q)), s(q))) \sim k}\; \vee : r}{\vdash f(\hat{S}(X_2(k, s(q)), s(q))) \nless s(k),\, f(X_2(k, s(q))) < k,\, f(\hat{S}(X_2(k, s(q)), s(q))) \sim k}\; \vee : r}{f(\hat{S}(X_2(k, s(q)), s(q))) < s(k) \vdash f(X_2(k, s(q))) < k,\, f(\hat{S}(X_2(k, s(q)), s(q))) \sim k}\; \neg : r$$

(2)

$$\dfrac{\begin{array}{c}(\delta_1, (n, m, p(w), s(k), s(m), 0))\\ \vdash f(Y(p(w), s(k))) < s(k)\end{array} \qquad (2)}{\vdash f(Y(w, k)) < k,\, f(\hat{S}(Y(w, k), s(q))) \sim k}\; \mathrm{Res}(\mu_1)$$

(1)

$$\dfrac{(1) \qquad \dfrac{\dfrac{\dfrac{\begin{array}{c}(\delta_2, (n, m, w, k, p(r), s(q)))\\ \vdash f(Y_1(w, k)) < k,\, \hat{F}_5(\mathbf{X}, k, s(q))\end{array}}{\vdash f(Y_1(w, k)) < k,\, f(X_3(k)) \nsim k \vee \hat{F}_5(\mathbf{X}, k, q)}\; S\hat{F}_5 r}{\vdash f(Y_1(w, k)) < k,\, f(X_3(k)) \nsim k,\, \hat{F}_5(\mathbf{X}, k, q)}\; \vee : r}{f(X_3(k)) \sim k \vdash f(Y_1(w, k)) < k,\, \hat{F}_5(\mathbf{X}, k, q)}\; \neg : r}{\vdash f(Y(w, k)) < k,\, \hat{F}_5(\mathbf{X}, k, q)}\; \mathrm{Res}(\mu_2)$$

where

$$\mu_1 = \left\{ X_2(k, s(q)) \leftarrow Y(w, k),\ Y(p(w), s(k)) \leftarrow \hat{S}(Y(w, k), s(q)) \right\},$$

and

$$\mu_2 = \left\{ Y_1(w, k) \leftarrow Y(w, k),\ X_3(k) \leftarrow \hat{S}(Y(w, k), s(q)) \right\}.$$

- $\rho_2(\delta_2, \mathbf{X}, Y, n, m, w, k, r, q)$

$$\begin{array}{cl}
\left(\delta_3, (n, m, w, k, 0, s(q))\right) & \\
\vdash \hat{F}_4(\mathbf{X}, k, s(q)) & \\ \hline
\vdash (f(\hat{S}(X_2(k, s(q)), s(q))) \nless s(k) \vee f(X_2(k, s(q))) < k \vee f(\hat{S}(X_2(k, s(q)), s(q))) \sim k) \wedge \hat{F}_4(\mathbf{X}, k, q) & S\hat{F}_4 r \\ \hline
\vdash f(\hat{S}(X_2(k, s(q)), s(q))) \nless s(k) \vee f(X_2(k, s(q))) < k \vee f(\hat{S}(X_2(k, s(q)), s(q))) \sim k & \wedge : r_2 \\ \hline
\vdash f(\hat{S}(X_2(k, s(q)), s(q))) \nless s(k), f(X_2(k, s(q))) < k \vee f(\hat{S}(X_2(k, s(q)), s(q))) \sim k & \vee : r \\ \hline
\vdash f(\hat{S}(X_2(k, s(q)), s(q))) \nless s(k), f(X_2(k, s(q))) < k, f(\hat{S}(X_2(k, s(q)), s(q))) \sim k & \vee : r \\ \hline
f(\hat{S}(X_2(k, s(q)), s(q))) < s(k) \vdash f(X_2(k, s(q))) < k, f(\hat{S}(X_2(k, s(q)), s(q))) \sim k & \neg : r \\
(2) &
\end{array}$$

$$\begin{array}{c}
\dfrac{\begin{array}{c}(\delta_1, (n, m, p(w), s(k), s(m), 0)) \\ \vdash f(Y(p(w), s(k))) < s(k)\end{array} \qquad (2)}{\vdash f(Y(w, k)) < k, f(\hat{S}(Y(w, k), s(q))) \sim k}\ \mathrm{Res}(\mu_1) \\
(1)
\end{array}$$

$$\dfrac{(1) \qquad \begin{array}{cl}
(\delta_5, (n, m, w, k, 0, s(q))) & \\
\vdash \hat{F}_3(\mathbf{X}, k, s(q)) & \\ \hline
\vdash \hat{F}_5(\mathbf{X}, k, s(q)) \wedge \hat{F}_4(\mathbf{X}, k, s(q)) \wedge \hat{F}_3(\mathbf{X}, p(k), s(q)) & S\hat{F}_3 r \\ \hline
\vdash \hat{F}_5(\mathbf{X}, k, s(q)) \wedge \hat{F}_4(\mathbf{X}, p(k), s(q)) & \wedge : r \\ \hline
\vdash \hat{F}_5(\mathbf{X}, k, s(q)) & \wedge : r \\ \hline
\vdash f(X_3(k)) \nsim k \vee \hat{F}_5(\mathbf{X}, k, q) & S\hat{F}_5 r \\ \hline
\vdash f(X_3(k)) \nsim k, \hat{F}_5(\mathbf{X}, k, q) & \vee : r \\ \hline
f(X_3(k)) \sim k \vdash \hat{F}_5(\mathbf{X}, k, q) & \neg : r
\end{array}}{\vdash f(Y(w, k)) < k, \hat{F}_5(\mathbf{X}, k, q)}\ \mathrm{Res}(\mu_2)$$

where

$$\mu_1 = \left\{ X_2(k, s(q)) \leftarrow Y(w, k),\ Y(p(w), s(k)) \leftarrow \hat{S}(Y(w, k), s(q)) \right\},$$

and

$$\mu_2 = \left\{ X_3(k) \leftarrow \hat{S}(Y(w, k), s(q)) \right\}.$$

– $\Pi_3(\delta_3, \mathbf{X}, Y, n, m, w, k, r, q) =$

$$\left\{ \begin{array}{l} (\delta_3, \mathbf{X}, Y, n, m, w, k, r, q) \rightarrow \rho_1(\delta_3, \mathbf{X}, Y, n, m, w, k, r, q) \colon r > 0 \\ (\delta_3, \mathbf{X}, Y, n, m, w, k, r, q) \rightarrow \rho_2(\delta_3, \mathbf{X}, Y, n, m, w, k, r, q) \colon r = 0 \end{array} \right\}.$$

- $\rho_1(\delta_3, \mathbf{X}, Y, n, m, w, k, r, q)$

$$\begin{array}{cl}
(\delta_3, (n, m, w, k, p(r), s(q))) & \\
\vdash \hat{F}_4(\mathbf{X}, k, s(q)) & \\ \hline
\vdash M \wedge \hat{F}_4(\mathbf{X}, k, q) & S\hat{F}_4 r \\ \hline
\vdash \hat{F}_4(\mathbf{X}, k, q) & \wedge : r
\end{array}$$

where M denotes

$$\begin{array}{l}(\neg f(\hat{S}(X_2(k, s(q)), s(q))) < s(k) \vee f(X_2(k, s(q))) < k \vee \\ \quad f(\hat{S}(X_2(k, s(q)), s(q))) \sim k).\end{array}$$

- $\rho_2(\delta_3, \mathbf{X}, Y, n, m, w, k, r, q)$

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}(\delta_5,(n,m,p(w),s(k),0,q))\\ \vdash \hat{F}_3(\mathbf{X},s(k),q)\end{array}}{\vdash \hat{F}_5(\mathbf{X},s(k),q)\wedge \hat{F}_4(\mathbf{X},k,q)\wedge \hat{F}_3(\mathbf{X},k,q)}\;S\hat{F}_3r}{\vdash \hat{F}_5(\mathbf{X},s(k),q)\wedge \hat{F}_4(\mathbf{X},k,q)}\;\wedge:r}{\vdash \hat{F}_4(\mathbf{X},k,q)}\;\wedge:r$$

– $\Pi_4(\delta_4,\mathbf{X},Y,n,m,w,k,r,q)=$

$$\left\{\begin{array}{l}(\delta_4,\mathbf{X},Y,n,m,w,k,r,q)\rightarrow\rho_1(\delta_4,\mathbf{X},Y,n,m,w,k,r,q)\colon r>0\\ (\delta_4,\mathbf{X},Y,n,m,w,k,r,q)\rightarrow\rho_2(\delta_4,\mathbf{X},Y,n,m,w,k,r,q)\colon r=0\end{array}\right\}.$$

- $\rho_1(\delta_4,\mathbf{X},Y,n,m,w,k,r,q)$

$$\dfrac{\dfrac{\begin{array}{c}(\delta_4,(n,m,w,k,p(r),s(q)))\\ \vdash \hat{F}_2(\mathbf{X},k,s(q))\end{array}}{\vdash (f(\hat{S}(X_1(k,s(q)),s(q)))\sim k\ \vee\ f(X_1(k,s(q)))<k)\wedge \hat{F}_2(\mathbf{X},k,q)}\;S\hat{F}_2r}{\vdash \hat{F}_2(\mathbf{X},k,q)}\;\wedge:r$$

- $\rho_2(\delta_4,\mathbf{X},Y,n,m,w,k,r,q)$

$$\dfrac{\dfrac{\begin{array}{c}\left(\delta_4',(n,m,w,k,r,q)\right)\\ \vdash \hat{F}_1(\mathbf{X},k,q)\end{array}}{\vdash \hat{F}_2(\mathbf{X},k,q)\wedge \hat{F}_3(\mathbf{X},k,q)}\;S\hat{F}_1r}{\vdash \hat{F}_2(\mathbf{X},k,q)}\;\wedge:r$$

– $\Pi_5(\delta_5,\mathbf{X},Y,n,m,w,k,r,q)=$

$$\left\{\begin{array}{l}(\delta_5,\mathbf{X},Y,n,m,w,k,r,q)\rightarrow\rho_1(\delta_5,\mathbf{X},Y,n,m,w,k,r,q)\colon w>0\\ (\delta_5,\mathbf{X},Y,n,m,w,k,r,q)\rightarrow\rho_2(\delta_5,\mathbf{X},Y,n,m,w,k,r,q)\colon w=0\end{array}\right\}.$$

- $\rho_1(\delta_5,\mathbf{X},Y,n,m,w,k,r,q)$

$$\dfrac{\dfrac{\begin{array}{c}(\delta_5,(n,m,p(w),s(k),r,q))\\ \vdash \hat{F}_3(\mathbf{X},s(k),q)\end{array}}{\vdash \hat{F}_5(\mathbf{X},s(k),q)\wedge \hat{F}_4(\mathbf{X},k,q)\wedge \hat{F}_3(\mathbf{X},k,q)}\;S\hat{F}_3r}{\vdash \hat{F}_3(\mathbf{X},k,q)}\;\wedge:r$$

- $\rho_2(\delta_5,\mathbf{X},Y,n,m,w,k,r,q)$

$$\dfrac{\dfrac{\begin{array}{c}\left(\delta_5',(n,m,w,k,r,q)\right)\\ \vdash \hat{F}_1(\mathbf{X},k,q)\end{array}}{\vdash \hat{F}_2(\mathbf{X},k,q)\wedge \hat{F}_3(\mathbf{X},k,q)}\;S\hat{F}_1r}{\vdash \hat{F}_3(\mathbf{X},k,q)}\;\wedge:r$$

– $\Pi_6(\delta_6,\mathbf{X},Y,n,m,w,k,r,q)=$

$$\left\{\begin{array}{l}(\delta_6,\mathbf{X},Y,n,m,w,k,r,q)\rightarrow\rho_1(\delta_6,\mathbf{X},Y,n,m,w,k,r,q)\colon w>0\\ (\delta_6,\mathbf{X},Y,n,m,w,k,r,q)\rightarrow\rho_2(\delta_6,\mathbf{X},Y,n,m,w,k,r,q)\colon w=0\end{array}\right\}.$$

- $\rho_1(\delta_6,\mathbf{X},Y,n,m,w,k,r,q)$ denotes

$$\begin{array}{cl}
(\delta_3, (n, m, w, k, p(r), 0)) & \\
\vdash \hat{F}_4(\mathbf{X}, k, 0) & \\ \hline
\vdash f(X_2(k, 0) \not< s(k) \vee f(X_2(k, 0)) < k \vee f(X_2(k, 0)) \sim k & B\hat{F}_4 r \\ \hline
\vdash f(X_2(k, 0)) \not< s(k), f(X_2(k, 0)) < k \vee f(X_2(k, 0)) \sim k & \vee : r \\ \hline
\vdash f(X_2(k, 0)) \not< s(k), f(X_2(k, 0)) < k, f(X_2(k, 0)) \sim k & \vee : r \\ \hline
f(X_2(k, 0)) < s(k) \vdash f(X_2(k, 0)) < k, f(X_2(k, 0)) \sim k & \neg : r \\
(2) &
\end{array}$$

$$\begin{array}{cl}
(\delta_1, (n, m, p(w), s(k), r, 0)) & \\
\vdash f(Y(p(w), s(k))) < s(k) \qquad (2) & \\ \hline
\vdash f(Y(p(w), s(k))) < k, f(Y(p(w), s(k))) \sim k & \mathrm{Res}(\mu_1) \\
(1) &
\end{array}$$

$$\dfrac{(1) \qquad \begin{array}{cl}
(\delta_5, (n, m, w, k, p(r), 0)) & \\
\vdash \hat{F}_3(\mathbf{X}, 0, m) & \\ \hline
\vdash \hat{F}_5(\mathbf{X}, k, 0) \wedge f(a) \not< 0 & B\hat{F}_3 r \\ \hline
\vdash \hat{F}_5(\mathbf{X}, k, 0) & \wedge : r_2 \\ \hline
\vdash f(\hat{S}(X_3(k), 0)) \nsim k & B\hat{F}_5 r \\ \hline
\vdash f(X_3(k)) \nsim k & B\hat{S} r \\ \hline
f(X_3(k)) \sim k \vdash & \neg : r
\end{array}}{\vdash f(Y(w, k)) < k}\ \mathrm{Res}(\mu_2)$$

where

$$\mu_1 = \left\{ X_2(k, 0) \leftarrow Y(p(w), s(k)) \right\},$$

and

$$\mu_2 = \left\{ Y(p(w), s(k)) \leftarrow Y(w, k)\ ,\ X_3(k) \leftarrow Y(w, k) \right\}.$$

- $\rho_2(\delta_6, \mathbf{X}, Yn, m, w, k, r, q)$

$$\dfrac{\begin{array}{cl}
(\delta_4, (n, m, w, k, p(r), 0)) & \\
\vdash \hat{F}_2(\mathbf{X}, k, 0) & \\ \hline
\vdash f(X_1(k, 0)) < k \vee f(X_1(k, 0)) \sim k & B\hat{F}_2 r \\ \hline
\vdash f(X_1(k, 0)) < k, f(X_1(k, 0)) \sim k & \vee : r \\ \hline
\vdash f(X_1(k, 0)) < k, f(X_1(k, 0)) \sim k & B\hat{S} r
\end{array} \qquad \begin{array}{cl}
(\delta_5, (n, m, w, k, p(r), 0)) & \\
\vdash \hat{F}_3(\mathbf{X}, k, 0) & \\ \hline
\vdash \hat{F}_5(\mathbf{X}, k, 0) \wedge f(a) \not< 0 & B\hat{F}_3 r \\ \hline
\vdash \hat{F}_5(\mathbf{X}, k, 0) & \wedge : r_2 \\ \hline
\vdash f(\hat{S}(X_3(k), 0)) \nsim k & B\hat{F}_5 r \\ \hline
\vdash f(X_3(k)) \nsim k & B\hat{S} r \\ \hline
f(X_3(k)) \sim k \vdash & \neg : r
\end{array}}{\vdash f(Y(w, k)) < k}\ \mathrm{Res}(\mu)$$

where $\mu = \left\{ X_1(k, 0) \leftarrow Y(w, k)\ ,\ X_3(k) \leftarrow Y(w, k) \right\}$.

In Example 23 we constructed an RPL_0^{Ψ} refutation $\mathcal{D}$. From this refutation we may extract a point transition system following the construction outlined in Definition 47.

Example 24 The point transition system for $\mathcal{D}$, as defined in Example 23, is $pts(\mathcal{D}) = (\delta_0, \Delta^*, \Delta_a, \mathbf{P})$, for $\mathbf{P} = \bigcup P_{\delta_i}$ where the P_{δ_i}'s are defined as follows:

- $P_{\delta_0} = pts(\Pi_0(\delta_0, \mathbf{X}, Y, n, m)) =$

$$\left\{ \begin{array}{c}
(\delta_0, (n, m)) \rightarrow \left\{ \begin{array}{c} (\delta_1, (n, m, n, n, s(m), 0)) \\ (\delta_5, (n, m, n, n, 0, m)) \end{array} \right\} : n > 0 \wedge m > 0 \\
(\delta_0, (n, m)) \rightarrow \left\{ \begin{array}{c} (\delta_1, (0, m, 0, 0, s(m), 0)) \\ (\delta_0^4, (0, m)) \end{array} \right\} : n = 0 \wedge m > 0 \\
(\delta_0, (n, m)) \rightarrow \left\{ \begin{array}{c} (\delta_5, (n, 0, n, n, 0, 0)), \\ (\delta_6, (n, 0, n, n, s(0), 0)) \end{array} \right\} : n > 0 \wedge m = 0 \\
(\delta_0, (n, m)) \rightarrow \left\{ \begin{array}{c} (\delta_0^1, (0, 0)) \\ (\delta_0^2, (0, 0)) \\ (\delta_0^3, (0, 0)) \end{array} \right\} : n = 0 \wedge m = 0
\end{array} \right\}$$

- $P_{\delta_1} = pts(\Pi_1(\delta_1, \mathbf{X}, Y, n, m, w, k, r, q)) =$

$$\left\{ \begin{array}{c}
(\delta_1, (n, m, w, k, r, q)) \rightarrow \left\{ \begin{array}{c} (\delta_1, (n, m, p(w), s(k), r, 0)), \\ (\delta_2, (n, m, w, k, p(p(r)), 0)), \\ (\delta_3, (n, m, w, k, p(r), 0)) \end{array} \right\} : w > 0 \\
(\delta_1, (n, m, w, k, r, q)) \rightarrow \left\{ \begin{array}{c} (\delta_4, (n, m, 0, k, p(r), 0)), \\ (\delta_2, (n, m, 0, k, p(p(r)), 0)) \end{array} \right\} : w = 0
\end{array} \right\}$$

- $P_{\delta_2} = pts(\Pi_2(\delta_2, \mathbf{X}, Y, n, m, w, k, r, q)) =$

$$\left\{ \begin{array}{c}
(\delta_2, (n, m, w, k, r, q)) \rightarrow \left\{ \begin{array}{c} (\delta_2, (n, m, w, k, p(r), s(q))), \\ (\delta_1, (n, m, p(w), s(k), s(m), 0)), \\ (\delta_3, (n, m, w, k, p(r), s(q))) \end{array} \right\} : r > 0 \\
(\delta_2, (n, m, w, k, r, q)) \rightarrow \left\{ \begin{array}{c} (\delta_1, n, m, p(w), s(k), s(m), 0), \\ (\delta_3, n, m, w, k, 0, s(q)), \\ (\delta_5, n, m, w, k, 0, s(q)) \end{array} \right\} : r = 0
\end{array} \right\}$$

- $P_{\delta_3} = pts(\Pi_3(\delta_3, \mathbf{X}, Y, n, m, w, k, r, q)) =$

$$\left\{ \begin{array}{c}
(\delta_3, (n, m, w, k, r, q)) \rightarrow \left\{ (\delta_3, n, m, w, k, p(r), s(q)), \right\} : r > 0 \\
(\delta_3, (n, m, w, k, r, q)) \rightarrow \left\{ (\delta_5, n, m, p(w), s(k), 0, q) \right\} : r = 0
\end{array} \right\}$$

- $P_{\delta_4} = pts(\Pi_4(\delta_4, \mathbf{X}, Y, n, m, w, k, r, q)) =$

$$\left\{ \begin{array}{c}
(\delta_4, (n, m, w, k, r, q)) \rightarrow \left\{ (\delta_4, n, m, w, k, p(r), s(q)), \right\} : r > 0 \\
(\delta_4, (n, m, w, k, r, q)) \rightarrow \left\{ (\delta_4', n, m, w, k, r, q) \right\} : r = 0
\end{array} \right\}$$

- $P_{\delta_5} = pts(\Pi_5(\delta_5, \mathbf{X}, Y, n, m, w, k, r, q)) =$

$$\left\{ \begin{array}{c}
(\delta_5, (n, m, w, k, r, q)) \rightarrow \left\{ (\delta_5, n, m, p(w), s(k), r, q), \right\} : w > 0 \\
(\delta_5, (n, m, w, k, r, q)) \rightarrow \left\{ (\delta_5', n, m, w, k, r, q) \right\} : w = 0
\end{array} \right\}$$

- $P_{\delta_6} = pts(\delta_6, \mathbf{X}, Y, n, m, w, k, r, q)) =$

$$\left\{ \begin{array}{c}
(\delta_6, n, m, w, k, r, q) \rightarrow \left\{ \begin{array}{c} (\delta_1, n, m, p(w), s(k), r, 0), \\ (\delta_5, n, m, w, k, p(r), 0), \\ (\delta_3, n, m, w, k, p(r), 0) \end{array} \right\} : w > 0 \\
(\delta_6, n, m, w, k, r, q) \rightarrow \left\{ \begin{array}{c} (\delta_4, n, m, 0, k, p(r), 0), \\ (\delta_5, n, m, 0, k, p(r), 0) \end{array} \right\} : w = 0
\end{array} \right\}$$

By inspection one can see that $pts(\mathcal{D})$ is in fact a point transition system.

Theorem 4 *Let Ψ be the theory in Example 6. Then $\mathcal{D}$ in Example 23 is a convergent refutation schema of Ψ.*

Proof That $\mathcal{D}$ is a refutation schema of Ψ is easy to see as $S(\delta_0, X, Y, n, m) = \vdash$. It remains to show that $\mathcal{D}$ is convergent. By the Theorems 1 and 2 it is sufficient to prove that *pts*($\mathcal{D}$) (see Example 24) is canonic. We have $\Delta^* = \{\delta_0, \ldots, \delta_6\} \cup \Delta_a$ for $\Delta_a = \{\delta_0^1, \ldots, \delta_4^1, \delta_4', \delta_5'\}$. According to Definition 39 we have to partition Δ^* into Δ_1, Δ_2 for defining the order $<_P$. We define $\Delta_2 = \{\delta_0, \ldots, \delta_6\}$ and $\Delta_1 = \Delta_a$. Then it is easy to check that *pts*($\mathcal{D}$) is canonic. □

7 Future Work and Applications

The initial intention of this research was to develop a schematic resolution calculus and thus allowing interactive proof analysis using CERES-like methods [5] in the presence of induction. More precisely, the resolution calculus introduced in this work will provide the basis for a schematic CERES method more expressive than the methods proposed in [11,14]. As already indicated, the key to proof analysis using CERES lies in the fact that it provides a bridge between automated deduction and proof theory. In the schematic setting a bridge has been provided [11,14], and the formalism presented here provides a setting to study automated theorem proving for schematic first-order logic.

Our recursive semantics (defined in Sect. 5) separates local resolution derivations from the global "shape" of the refutation, an essential characteristic of induction. While constructing a recursive resolution refutation for a recursive unsatisfiable formula schema is incomplete, it is not clear whether the problem remains incomplete when the point transition system is fixed. In other words, we may instead ask: *"Is providing a recursive resolution refutation, with respect to a given point transition system, for recursive formulas complete?"* The answer to this question is not so clear in that it depends on the resolution calculus itself as well as on the associated unification problem. Both concepts are developed in this paper.

Concerning the resolution calculus presented in Sect. 4, both the Andrew's calculus-like sequent rules and the introduction of global variables provide the necessary extensions to resolution accommodating the recursive nature of our formulas. The unification problem discussed in Sect. 3 has not been addressed so far, and furthermore it may have interesting decidable fragments impacting schematic proof analysis as well as other fields.

Overall, the avenues we leave for future investigations provide ample opportunities for studying schematic theorem proving.

Acknowledgements This work was partially supported by the Linz Institute of Technology (LIT) Math$_{LP}$ project (LIT- 2019-7-YOU-213) funded by the state of upper Austria.

Funding Open access funding provided by TU Wien (TUW).

References

1. Andrews, P.B.: Resolution in type theory. J. Symb. Log. **36**(3), 414–432 (1971)
2. Aravantinos, V., Caferra, R., Peltier, N.: Decidability and undecidability results for propositional schemata. J. Artif. Intell. Res. **40**(1), 599–656 (2011)
3. Aravantinos, V., Mnacho, E., Nicolas, P.: A resolution calculus for first-order schemata. Fundam. Inform. **125**, 101–133 (2013)
4. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press, Cambridge (1998)
5. Baaz, M., Hetzl, S., Leitsch, A., Richter, C., Spohr, H.: Ceres: an analysis of Fürstenberg's proof of the infinity of primes. Theor. Comput. Sci. **403**(2–3), 160–175 (2008)
6. Baaz, M., Leitsch, A.: Cut-elimination and redundancy-elimination by resolution. J. Symb. Comput. **29**, 149–176 (2000)
7. Baaz, M., Leitsch, A.: Methods of Cut-elimination, vol. 34. Springer, Berlin (2011)
8. Brotherston, J.: Cyclic proofs for first-order logic with inductive definitions. In: TABLEAUX. Lecture Notes in Computer Science, vol. 3702, pp. 78–92. Springer, Berlin (2005)
9. Brotherston, J., Simpson, A.: Sequent calculi for induction and infinite descent. J. Logic Comput. **21**(6), 1177–1216 (2010)
10. Cerna, D.M., Leitsch, A.: Schematic cut elimination and the ordered pigeonhole principle. In: Automated Reasoning—8th International Joint Conference, IJCAR 2016, Coimbra, Portugal, June 27–July 2, 2016, Proceedings, pp. 241–256 (2016). Peer Reviewed
11. Dunchev, C., Leitsch, A., Rukhaia, M., Weller, D.: Cut-elimination and proof schemata. In: TbiLLC. Lecture Notes in Computer Science, vol. 8984 , pp. 117–136. Springer (2013)
12. Hetzl, S., Leitsch, A., Weller, D., Paleo, B.W.: Herbrand sequent extraction. In: Intelligent Computer Mathematics, pp. 462–477. Springer (2008)
13. Kersani, A.: Preuves par induction dans le calcul de superposition. (Induction proof in superposition calculus). PhD thesis, Grenoble Alpes University, France (2014)
14. Leitsch, A., Peltier, N., Weller, D.: CERES for first-order schemata. J. Log. Comput. **27**(7), 1897–1954 (2017)
15. Mcdowell, R.: Cut-elimination for a logic with definitions and induction. Theor. Comput. Sci. **232**, 91–119 (1997)
16. Simpson, S.G.: Subsystems of Second Order Arithmetic. Perspectives in Mathematical Logic. Springer, Berlin (1999)
17. Takeuti, G.: Proof Theory. Studies in Logic and the Foundations of Mathematics, vol. 81. American Elsevier Pub., New York (1975)