A Resolution Calculus for Recursive Clause Sets: Extended Version

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

[1] Research Institute for Symbolic Computation (RISC), Johannes Kepler University, Linz, Austria
David.Cerna@risc.jku.at

[2] Institute of Logic and Computation, Theory and Logic Group, TU Wien, Vienna, Austria
leitsch@logic.at

[3] Institute of Discrete Mathematics and Geometry, TU Wien, Vienna, Austria
anela@logic.at

Abstract. Proof schemata provide an alternative formalism for proofs containing an inductive argument, which allow an extension of *Herbrand's theorem* and thus, the construction of *Herbrand sequents* and *expansion trees*. Existing proof transformation methods for proof schemata rely on constructing a recursive resolution refutation, a task which is highly non-trivial. An automated method for constructing such refutations exists, but only works for a very weak fragment of arithmetic and is hard to use interactively. In this paper we introduce a simplified schematic resolution calculus, based on *definitional clause forms* allowing interactive construction of refutations beyond existing automated methods. Furthermore we provide a refutation of a recursive clause set, which cannot be finitely represented in existing formalisms.

1 Introduction

Proof schemata serve as an alternative formulation of induction through primitive recursive proof specification. Prior to the formalization of the concept, an analysis of Fürstenberg's proof of the infinitude of primes [4] suggested the need for a formalism for inductive proofs that allows proof analysis. The underlying method for this analysis was **CERES** [5] (cut-elimination by resolution) which, unlike reductive cut-elimination, can be applied to recursively defined proofs by extracting a *schematic characteristic formula* and constructing a recursively defined refutation. As the name reveals, the construction of a recursively defined refutation forms the heart of the method, which, unfortunately, is a highly complex task. Existing methods are either incomplete [15] or only work for a weak fragment of arithmetic and are hard to use interactively [19].

Historically, it was the schematic construction of object level concepts that formed the basis for proof schemata. These foundational ideas evolved from work

*This research is supported by FWF project P28789-N32

by V. Aravantinos et al. [1, 2]. Initially, they considered formulas of an indexed propositional logic with a single *free numeric parameter* and with two new logical connectors, i.e. $\vee$-iteration and $\wedge$-iteration. Their investigations led to a tableau-based decision procedure for the satisfiability of a monadic[1] fragment of this logic. An extension to a special case of multiple parameters was also investigated by D. Cerna [9]. In a more recent work, V. Aravantinos et al. [2] introduced a superposition resolution calculus for a clausal representation of indexed propositional logic. The calculus yielded decidability results for an even larger fragment than the monadic one. The clausal form allows an easy extension to indexed predicate logic, see [19].

These results inspired investigations into the use of schemata as an alternative formalization of induction for proof analysis and proof transformation. Note that this is not the first alternative formalization of induction with respect to Peano arithmetic [23]. However, all existing examples [7, 8, 20], to the best of our knowledge, lack a proof normal form or *subformula-like property*. Subformula property means that every formula occurring in the proof is a subformula of a formula occurring in the end-sequent. By the term "subformula-like", we mean that the proof is not necessarily fully analytic [2], but may contain quantifier-free cuts. In this sense cut-elimination in the presence of induction results in a non-analytic proof: some part of the argument is not directly connected to the theorem being proven. Two important constructions extractable from proofs with subformula-like properties, *Herbrand sequents* [17, 23] and *expansion trees* [21], cannot directly be obtained from proofs within these calculi. While Herbrand sequents allow the representation of the propositional content of first-order proofs, expansion trees generalize Herbrand's theorem. Note that in [19] finite representations of 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 of Herbrand systems does result in sequents derivable from a finite set of ground instances. In some cases, the resulting Herbrand system is not formalizable as a proof schema in the sense of [14, 19] due to the restriction placed on proof schema construction, see [10].

The formalism developed in this paper extends and improves the formal framework for schematic cut-elimination defined in [15]. In [15] Herbrand systems could be constructed, but the method is incomplete and a schematic representation of normal forms could not be obtained. Despite these defects the method could be used for analyzing the pigeonhole principle [11]. An improved version of this cut-elimination method has been introduced in [19], using the superposition resolution calculus of [2]. The method is complete and always produces normal forms; but it is less expressive than the former one defined in [15]. Note that the method of [15] can formalize proof normal forms with a non-elementary length with respect to the size of the end-sequent[3].

[1] In this fragment the use of schematic constructors is restricted to one free parameter per formula.

[2] A proof is analytic when it fulfills the subformula property.

[3] See Orevkov's proof [22] or Boolos' proof [6].

In this work we present a schematic resolution calculus which improves the calculus of [15], in two ways: first, it simplifies the formalism considerably (by using schematic definitional clause normal forms in resolution), and second it extends the power of resolution deduction schemata by admitting more than just one parameter. Moreover, the method improves the approach in [19] by increasing its strength considerably and by providing a simple and elegant formalism for interactive use. In contrast to [15], unification is locally applied and linking of resolution derivations is simplified, similar to [12, 19]. Furthermore, our calculus can be used to construct a normal form and hence for proof analysis. Due to the locally applied unification and the use of so-called globalization variables, we can generalize a method for efficient proof mining, see [18], and omit the construction of normal forms by extracting Herbrand sequents and expansion trees from *proof projections* (cut-free proofs obtained by the method CERES).

We believe that the simple and powerful schematic framework defined in this paper will provide a valuable tool for inductive proof mining by cut-elimination in the future.

2 Preliminaries

Covering proof schemata in detail is not possible given the space constraints, nor is it necessary as we only need a few facets of the foundational theory for this work. For a more thorough introduction to proof schemata we refer the reader to the following papers [10–13, 15, 19].

We work in a two-sorted version of classical first-order logic. The first sort we consider is ω, in which every term normalizes to a *numeral*, i.e. a term inductively constructable by $N \Rightarrow s(N) \mid 0$, such that $s(N) \neq 0$ and $s(N) = s(N') \rightarrow N = N'$. Numerals will be denoted by lower-case Greek letters (α, β, γ, etc). Furthermore, the ω sort includes a countable set of parameter symbols $\mathcal{N}$. Parameters will be denoted by n, m.

The second sort, the ι-sort (individuals), is a standard first-order term language extended by *defined function symbols* and *schematic variable symbols*. Defined function symbols, i.e. primitive recursively defined functions, will be denoted with $\hat{\cdot}$. Schematic variable symbols are variables of the type $\omega \rightarrow \iota$ and are used to construct sequences of variables. Given a schematic variable x instantiated by a numeral α we obtain a variable of the ι-sort $x(\alpha)$.

Formula schemata, a generalization of formulas including *defined predicate symbols*, are defined inductively using the standard logical connectives from uniterpreted and defined predicate symbols. Defined predicate symbols will be denoted with $\hat{\cdot}$.

A *schematic sequent* is a pair of two multisets of formula schemata Δ, Π denoted by $\Delta \vdash \Pi$. Multisets of formula schemata will be denoted by upper-case Greek letters unless it causes confusion.

We extend the LK-calculus to the LKE-calculus by adding an inference rule for the construction of defined predicate and function symbols and a set of convergent rewrite rules $\mathcal{E}$ (equational theory). The underlying equations of $\mathcal{E}$

are of the form $\widehat{f}(\bar{t}) = E$, where $\bar{t}$ contains no defined symbols, and either $\widehat{f}$ is a defined function symbol and E is a term or $\widehat{f}$ is a defined predicate symbol and E is a formula schema.

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

The schematic CERES method is based on the extraction of a schematic characteristic formula, which is always unsatisfiable [15, 19]. Constructing a schematic resolution refutation of the schematic characteristic formula forms the heart of the method. Having obtained a resolution refutation, it is possible to construct a schematic normal form and extract Herbrand systems (a generalization of Herbrand's theorem).

3 Schematic Resolution Deductions

In this section we define a resolution calculus for schematic clause sets. This calculus differs from a former schematic clause calculus defined in [15] in various ways: 1. the new calculus presented here does not need the concept of clause schemata and clause set schemata, 2. there is no need to separate unification from the specification of resolution deductions, and 3. the calculus is more powerful by admitting a second parameter in the definition of recursions. Moreover the new calculus is much more powerful than the superposition calculus used in [19] for schematic cut-elimination. Our approach here is based on the method of structural clause form transformation (we use a variant of the p-definitional form [3]) which is well suited for treating recursively defined formulas. As the refutational problems in schematic CERES are defined via negation normal forms (NNF) we focus on the refutation of universal NNF schemata.

Definition 2 (definitional clause set). *Let F be a formula in NNF. We define a set of clauses $\Phi(F, L_F)$ (the definitional clause set of F) where L_F is an atom labelled by F; we proceed by induction on the complexity of F.*

- *F is an atom over the variables $\overline{y}$ (where $\overline{y}$ is a tuple of variables of type i). Then we define*
$$\Phi(F, L_F) = \{L_F(\overline{y}) \vdash F\}.$$
- *$F = \neg A$ where A is an atom over $\overline{y}$. Then*
$$\Phi(F, L_F) = \{L_F(\overline{y}), A \vdash\}.$$
- *$F = F_1 \wedge F_2$. Assume that $\Phi(F_1, L_{F_1}), \Phi(F_2, L_{F_2})$ are defined and that $L_{F_1}(\overline{y}_1), L_{F_2}(\overline{y}_2)$ are atoms over $\overline{y}$. Then $\Phi(F, L_F)$ is defined as*
$$\{L_F(\overline{y}) \vdash L_{F_1}(\overline{y}_1) \quad , \quad L_F(\overline{y}) \vdash L_{F_2}(\overline{y}_2))\} \cup \Phi(F_1, L_{F_1}) \cup \Phi(F_2, L_{F_2}).$$
- *$F = F_1 \vee F_2$. Analogous with*
$$\Phi(F, L_F) = \{L_F(\overline{y}) \vdash L_{F_1}(\overline{y}_1), L_{F_2}(\overline{y}_2)\} \cup \Phi(F_1, L_{F_1}) \cup \Phi(F_2, L_{F_2}).$$

Definition 3 (definitional form). *Let $\Phi(F, L_F)$ be the definitional clause set of F where F is a formula in NNF over the variables $\overline{y}$. Then the set of clauses*

$$\mathrm{CL}(F, L_F)\colon \{\vdash L_F(\overline{y})\} \cup \Phi(F, L_F)$$

is called the definitional form *of F.*

Proposition 1. *Let F be a formula in NNF. Then F and $\mathrm{CL}(F, L_F)$ are sat-equivalent.*

Proof (sketch). Assume that F is satisfiable by a model $\mathcal{M}$. Then $\mathcal{M}$ can be extended to a model of $\mathrm{CL}(F, L_F)$ by assigning appropriate truth values to the defined atoms (this can easily be proved by induction on the logical complexity of F).

If F is unsatisfiable then we show that the standard clause form $\mathcal{C}(F)$ can be derived from $\mathrm{CL}(F, L_F)$ by resolution. We know that $\mathcal{C}(F)$ is unsatisfiable and thus, by $\mathrm{CL}(F, L_F) \models \mathcal{C}(F)$, also $\mathrm{CL}(F, L_F)$ is unsatisfiable.

We proceed by induction on the complexity of F. If F is an atom A over the variables $\overline{y}$ then $\vdash A$ is derivable from $\vdash L_A(\overline{y})$ and $L_A(\overline{y}) \vdash A$ by one resolution step. The case that $F = \neg A$ is analogous.

Let $F = F_1 \wedge F_2$ and assume that $\mathcal{C}(F_1)$ ($\mathcal{C}(F_2)$) is derivable via a resolution deduction ρ_1 (ρ_2) from $\mathrm{CL}(F_1, L_{F_1})$ ($\mathrm{CL}(F_2, L_{F_2})$). We know that $\mathcal{C}(F) = \mathcal{C}(F_1) \cup \mathcal{C}(F_2)$. So any refutation of $\mathcal{C}(F)$ can be transformed into a refutation of $\mathrm{CL}(F_1, L_{F_1}) \cup \mathrm{CL}(F_2, L_{F_2})$. But $\mathrm{CL}(F_1, L_{F_1}) \cup \mathrm{CL}(F_2, L_{F_2})$ can be derived from $\mathrm{CL}(F, L_F)$ by resolution.

The case $F = F_1 \vee F_2$ is slightly more complicated and can be solved by substituting two derivations into each other.

We can lift the definitions to the schematic case.

Definition 4 (schematic clause form). *Let Ψ be a schematic NNF definition with defined symbols $\hat{F}_1, \ldots, \hat{F}_k$ s.t. $\hat{F}_1 > \hat{F}_2 > \cdots > \hat{F}_k$ s.t.*

$$\hat{F}_i(\overline{y}_i, 0) = G_i(\overline{y}_i), \qquad \hat{F}_i(\overline{y}_i, n+1) = H_i(\overline{y}_i, F_i(\overline{y}_i, n)),$$

where the $G_i(\overline{y}_i)$ and $H_i(\overline{y}_i, \Diamond)$ are formulas in NNF not containing F_j for $j \leq i$. Then the structural form $\mathrm{CL}(\Psi)$ of Ψ is defined as $(\mathrm{CL}(\Psi, 0), \mathrm{CL}(\Psi, n+1))$ for

$$\begin{aligned} \mathrm{CL}(\Psi, 0) &= \{L_{F_1}(\overline{y}, 0)\} \cup \Phi(G_1(\overline{y}_1), L_{F_1}) \cup \cdots \cup \Phi(G_k(\overline{y}_k), L_{F_k}), \\ \mathrm{CL}(\Psi, n+1) &= \{L_{F_1}(\overline{y}, n+1)\} \cup \Phi(H_1(\overline{y}_1, L_{F_1}(\overline{y}_1, n)), L_{F_1}) \cup \cdots \cup \\ &\qquad \Phi(H_k(\overline{y}_k, L_{F_k}(\overline{y}_k, n)), L_{F_k}). \end{aligned}$$

where $\overline{y} = \overline{y}_1 \cdots \overline{y}_k$.

Note that, for any schematic NNF-definition Ψ, $\mathrm{CL}(\Psi)\colon (\mathrm{CL}(\Psi, 0), \mathrm{CL}(\Psi, n+1))$ is always a pair of finite sets of clauses.

Example 1. Consider the schematic formula $P(a) \wedge (\neg P(x) \vee P(f(x))) \wedge \neg P(f^n(a))$, a characteristic formula (in fact a simplified version under tautology elimination and subsumption) resulting from a schematic cut-elimination problem (see [19], Example 3.6) . The corresponding clause set schema is $\mathcal{C}_n = \{P(a), \neg P(x) \vee P(f(x)), \neg P(f^n(a))\}$ with the following schematic NNF formalization $\Psi =$

$$\begin{array}{ll}
\{\hat{F}(0,x) & = P(a) \\
\hat{F}(n+1,x) & = P(a) \wedge (\neg P(x) \vee P(f(x))) \\
\hat{G}(n) & = \neg P(\hat{f}(n,a)) \\
\hat{H}(0,x) & = \hat{F}(0,x) \wedge \hat{G}(0) \\
\hat{H}(n+1,x) & = \hat{F}(n+1,x) \wedge \hat{G}(n+1)\} \\
R(\hat{f}) & = \hat{f}(0,x) \rightarrow x \\
\hat{f}(n+1,x) \rightarrow f(\hat{f}(n,x)) &
\end{array}$$

with $\hat{H} > \hat{F}$, $\hat{H} > \hat{G}$. The structural form is $\mathrm{CL}(\Psi) = (\mathrm{CL}(\Psi, 0), \mathrm{CL}(\Psi, n+1))$, where

$$\begin{aligned}
\mathrm{CL}(\Psi, 0) = {} & \{\vdash L_H(0,x)\} \cup \{L_H(0,x) \vdash L_F(0,x);\ L_H(0,x) \vdash L_G(0); \\
& L_F(0,x) \vdash P(a);\quad L_G(0), P(\hat{f}(0,a)) \vdash\} \\
\mathrm{CL}(\Psi, n+1) = {} & \{\vdash L_H(n+1,x)\} \cup \{L_H(n+1,x) \vdash L_F(n+1,x); \\
& L_H(n+1,x) \vdash L_G(n+1);\ L_G(n+1), P(\hat{f}(n+1,a) \vdash; \\
& L_F(n+1,x) \vdash P(a);\quad L_F(n+1,x), P(x) \vdash P(f(x))\}.
\end{aligned}$$

Below we are defining a resolution calculus for schematic clause sets. A schematic clause is just an atomic sequent consisting of schematic atoms where schematically defined terms may occur.

Definition 5 (schematic clause). *Let $C\colon P_1, \ldots, P_\alpha \vdash Q_1, \ldots, Q_\beta$ be a sequent where the P_i and Q_j are schematic atoms and there are at most two parameters occurring in C. Then C is called a schematic clause.*

Note that this formalism is much simpler than that in [15] where schematic clauses required additional recursive definitions on the clause level. Having defined schematic clauses, we can define resolution deductions. Note that the concept of resolution deduction almost coincides with the usual one, with the only exception that the unification principle is defined over schematic terms and is undecidable (see [19]). Hence our definitions strongly differ from those in [15] where resolution terms were used.

Definition 6 (resolution deduction). *Let $\mathcal{C}$ be a set of schematic clauses over the parameters n, m. Then, for every clause $C \in \mathcal{C}$, C' (where C' is a variant of C) is a resolution deduction of C from $\mathcal{C}$. Assume that δ_1 is a resolution deduction of $C_1\colon \Gamma \vdash \Delta, A_1, \ldots, A_\alpha$ from $\mathcal{C}$ and δ_2 is a resolution deduction of $C_2\colon B_1, \ldots, B_\beta\, \Pi \vdash \Lambda$ from $\mathcal{C}$ s.t. δ_1 and δ_2 are variable disjoint (but* *both may contain the parameters n, m). Assume further that Θ is a unifier of $\{A_1, \ldots, A_\alpha, B_1, \ldots, B_\beta\}$. Then*

$$\frac{\begin{array}{cc}(\delta_1) & (\delta_2) \\ C_1 & C_2\end{array}}{C\colon \Gamma\Theta, \Pi\Theta \vdash \Delta\Theta, \Lambda\Theta}\ \Theta$$

is a resolution deduction of C from $\mathcal{C}$.

For schematic resolution deductions we need inductive definitions. Note that for schematic clause sets $\mathcal{C}$ over the parameter n we may deduce schematic clauses $D(m)$ where m is a parameter different from n. Therefore the concept of schematic resolution deduction requires two parameters in general. For defining proof schemata we introduce an infinite set of proof symbols Δ^*[4] and define a partial order $<$ on them.

Definition 7 (schematic resolution deduction). *Let Δ be a finite subset of Δ^* s.t. there exists a $\delta_0 \in \Delta$ with $\delta_0 > \delta'$ for all $\delta' \in \Delta$ s.t. $\delta' \neq \delta_0$. A finite set of tuples $\mathcal{R}\colon \{(\delta, \rho(\delta, n, 0), \rho(\delta, n, m{+}1)) \mid \delta \in \Delta\}$ is called a* schematic resolution deduction *from a finite set of schematic clauses $\mathcal{C}$ over the parameter n if the following conditions hold for every $\delta \in \Delta$:*
There exists a (possibly empty) finite set of clauses $\mathcal{C}(\delta)$ and a clause $D(\delta)$ s.t.
(1) $\rho(\delta, n, 0)$ is a resolution deduction of $D(\delta)\{m \leftarrow 0\}$ from $\mathcal{C} \cup \mathcal{C}(\delta)$,
(2) $\rho(\delta, n, m+1)$ is a resolution deduction of $D(\delta)\{m \leftarrow m+1\}$ from $\{D(\delta)\} \cup \mathcal{C} \cup \mathcal{C}(\delta)$,
(3) for all $C \in \mathcal{C}(\delta)$, C is a parameter instance (of the form $\{m \leftarrow t[n]\}$ for some term t over n) of a clause $D(\delta')$ for a $\delta' \in \Delta$ with $\delta > \delta'$ s.t. the conditions (1) and (2) hold for δ' .
If $D(\delta_0)$ is the empty clause $\vdash$ we call $\mathcal{R}$ a resolution refutation schema *of $\mathcal{C}$.*

Note that the concept is well defined as there are minimal elements in Δ for which $\mathcal{C}(\delta) = \emptyset$. A resolution deduction can be considered as a special case of a resolution deduction schema by only considering the base case and setting $\mathcal{C}(\delta) = \emptyset$.

The definition yields, for $\mathcal{C}(\delta) = \emptyset$, an algorithm which interprets $\rho(\delta, n, m)$ by constructing a resolution derivation of $D(\delta)\{m \leftarrow \beta\}$ from $\mathcal{C}\{n \leftarrow \gamma\}$ for any numerals β, γ replacing the parameters m, n. For $\mathcal{C}(\delta) \neq \emptyset$ we proceed inductively by interpreting the $\rho(\delta', n, m)$ for $\delta > \delta'$ inductively and replacing the clauses $D(\delta')\{m \leftarrow t[n]\}$ by the deductions $\rho(\delta', n, t[n])$.

Definition 8. *Let Ψ be a schematic NNF-definition and $\mathrm{CL}(\Psi, 0)$, $\mathrm{CL}(\Psi, n+1)$ be the corresponding set of clauses. Then the pair $(\mathcal{R}_b, \mathcal{R}_s)$, where $\mathcal{R}_b$ is a resolution refutation schema of $\mathrm{CL}(\Psi, 0)$ and $\mathcal{R}_s$ is a resolution refutation schema of $\mathrm{CL}(\Psi, n+1)$, is called a* refutation schema *for Ψ.*

[4] A proof symbol is a symbol denoting a proof.

Example 2. Consider the schematic NNF formalization Ψ and its structural clause form from Example 1, where

$$\begin{aligned}\mathrm{CL}(\Psi, 0) = \{\vdash L_H(0,x)\} \cup \{&L_H(0,x) \vdash L_F(0,x);\ L_H(0,x) \vdash L_G(0);\\ &L_F(0,x) \vdash P(a);\quad L_G(0), P(\hat{f}(0,a)) \vdash\}\end{aligned}$$

$$\begin{aligned}\mathrm{CL}(\Psi, n+1) = \{\vdash L_H(n+1,x)\} \cup \{&L_H(n+1,x) \vdash L_F(n+1,x);\\ &L_H(n+1,x) \vdash L_G(n+1);\ L_G(n+1), P(\hat{f}(n+1,a) \vdash;\\ &L_F(n+1,x) \vdash P(a);\quad L_F(n+1,x), P(x) \vdash P(f(x))\}.\end{aligned}$$

We define a refutation schema $(\mathcal{R}_b, \mathcal{R}_s)$ of Ψ.
Let $\mathcal{R}_b = \{(\delta^b, \rho(\delta^b, n, 0), \rho(\delta^b, n, m+1))\}$; we define $\rho(\delta^b, n, 0) = \rho(\delta^b, n, m+1)$ as follows:

Let ϕ_1 be

$$\dfrac{L_G(0), P(\hat{f}(0,a)) \vdash \qquad \dfrac{\vdash L_H(0,x) \quad L_H(0,y) \vdash L_G(0)}{\vdash L_G(0)}\ y \leftarrow x}{P(\hat{f}(0,a)) \vdash}\ \emptyset$$

and ϕ_2 be

$$\dfrac{\dfrac{\vdash L_H(0,u) \quad L_H(0,v) \vdash L_F(0,v)}{\vdash L_F(0,u)}\ v \leftarrow u \qquad L_F(0,w) \vdash P(a)}{\vdash P(a)}\ w \leftarrow u$$

Then $\rho(\delta^b, n, 0)$ is defined as

$$\dfrac{\phi_1 \quad \phi_2}{\vdash}\ \emptyset$$

Clearly $\rho(\delta^b, n, m)$ is an resolution refutation schema of $\mathrm{CL}(\Psi, 0)$.
Now we construct a schematic resolution derivation $\mathcal{R}_1 = \{(\delta, \rho(\delta, n, 0), \rho(\delta, n, m+1))\}$ where $D(\delta) = P(\hat{f}(m,a))$.
$\rho(\delta, n, 0) =$

$$\dfrac{\dfrac{\vdash L_H(n+1,y) \quad L_H(n+1,x) \vdash L_F(n+1,x)}{\vdash L_F(n+1,x)}\ y \leftarrow x \qquad L_F(n+1,z) \vdash P(a)}{\vdash P(a)}\ z \leftarrow x$$

Note that $D(\delta)\{m \leftarrow 0\} = P(a)$. $\rho(\delta, n, m+1) =$

$$\dfrac{\vdash P(\hat{f}(m,a)) \qquad \dfrac{\dfrac{\vdash L_H(n+1,y) \quad L_H(n+1,x) \vdash L_F(n+1,x)}{\vdash L_F(n+1,x)}\ y \leftarrow x \qquad C(z,n,m)}{P(x) \vdash P(f(x))}\ z \leftarrow x}{\vdash P(\hat{f}(m+1,a)}\ x \leftarrow \hat{f}(m,a)$$

for $C(z,n,m) = L_F(n+1,z), P(z) \vdash P(f(z))$.

$\rho(\delta, n, m+1)$ is a derivation of $D(\delta)\{m \leftarrow m+1\}$ from $\mathrm{CL}(\Psi, n+1) \cup \{D(\delta)\}$.

Finally we define the resolution refutation schema

$$\mathcal{R}_s\colon \{(\delta^s, \rho(\delta^s, n, 0), \rho(\delta^s, n, m+1)\} \cup \mathcal{R}_1$$

where $\delta^s > \delta$ and $\rho(\delta^s, n, 0) = \rho(\delta^s, n, m+1) =$

$$\dfrac{D(\delta)\{m \leftarrow n+1\} \qquad \dfrac{\dfrac{\vdash L_H(n+1,x) \qquad L_H(n+1,y) \vdash L_G(n+1)}{L_G(n+1)}\; y \leftarrow x \qquad \Gamma \vdash}{P(\hat{f}(n+1,a)) \vdash}\; \emptyset}{\vdash}\; \emptyset$$

where $\Gamma = L_G(n+1), P(\hat{f}(n+1,a))$. We have shown that $\mathcal{R}_s$ is a refutation schema of $\mathrm{CL}(\Psi, n+1)$ and therefore $(\mathcal{R}_b, \mathcal{R}_s)$ is a refutation schema of Ψ.

When we apply a total unification θ to a resolution refutation ρ of $\mathcal{C}$ then $\rho\theta$ becomes an LK-refutation of instances of $\mathcal{C}$ and thus of the universal closure of the corresponding formula. For defining the total unifier of a whole resolution deduction schema we first rename all resolution deductions in the definition recursively and then extend their total unifiers. To this aim we introduce for every δ sets of variables, so-called *globalization variables*, $X_i^\delta(0)$ and $X_i^\delta(m+1)$. We need these variables of type $\mathbb{N} \to \iota$ to avoid conflicts in variable substitutions in substituting the parameter m by a numeral.

Definition 9 (unification schema). *Let $\mathcal{R}$ be a resolution deduction schema from $\mathcal{C}$. We define sets of substitutions (unification schemata) $\Theta(\delta, n, 0)$ and $\Theta(\delta, n, m+1)$ for all $\delta \in \Delta$. We define the sets inductively beginning with the minimal $\delta \in \Delta$.*

– *Let δ be a minimal element in Δ. Then $\mathcal{C}(\delta) = \emptyset$. In this case $\rho(\delta, n, 0)$ is a resolution deduction of $D(\delta)\{m \leftarrow 0\}$ from $\mathcal{C}$ and $\rho(\delta, n, m+1)$ is a resolution deduction of $D(\delta)\{m \leftarrow m+1\}$ from $\mathcal{C} \cup \{D(\gamma)\}$.*

  Let $x_1, \ldots, x_\alpha$ be the variables occurring $\rho(\delta, n, 0)$. We define

  $$\eta(\delta, n, 0) = \{x_1 \leftarrow X_1^\delta(0), \ldots, x_\alpha \leftarrow X_\alpha^\delta(0)\}.$$

  Let $\theta(\delta, n, 0)$ be a total unifier of $\rho(\delta, n, 0)\eta(\delta, n, 0)$. Then we define $\Theta(\delta, n, 0) = \{\theta(\delta, n, 0)\}$.

  Assume that $\rho(\delta, n, m+1)$ contains the variables $y_1, \ldots, y_\beta$. We define

  $$\eta(\delta, n, m+1) = \{y_1 \leftarrow X_1^\delta(m+1), \ldots, y_\beta \leftarrow X_\beta^\delta(m+1)\}.$$

  Let $\theta(\delta, n, m+1)$ be a total unifier of $\rho(\delta, n, m+1)\eta(\delta, n, m+1)$. Then

  $$\Theta(\delta, n, m+1) = \Theta(\delta, n, m) \cup \{\theta(\delta, n, m+1)\}.$$

– *Let δ be a non-minimal element in Δ and $\mathcal{C}(\delta) \neq \emptyset$. Let*

$$\mathcal{C}(\delta) = \{D(\delta_1)\{m \leftarrow t_1[n]\}, \ldots, D(\delta_l)\{m \leftarrow t_l[n]\}\}$$

where $\delta > \delta_j$ for $j = 1, \ldots, l$. Let $\lambda(\delta, n, 0)$ and $\eta(\delta, n, 0)$ as above and $\theta(\delta, n, 0)$ be a total unifier of $\rho(\delta, n, 0)\eta(\delta, n, 0)$. Then

$$\Theta(\delta, n, 0) = \{\theta(\delta, n, 0)\} \cup \Theta(\delta_1, n, t_1[n]])\eta(\delta, n, 0)\theta(\delta, n, 0) \cup \ldots$$
$$\ldots \cup \Theta(\delta_l, n, t_l[n])\eta(\delta, n, 0)\theta(\delta, n, 0).$$

Similarly we obtain

$$\begin{aligned}\Theta(\delta, n, m+1) &= \Theta(\delta, n, m) \cup \{\theta(\delta, n, m+1)\}\\ &\cup \Theta(\delta_1, n, t_1[n])\eta(\delta, n, m+1)\theta(\delta, n, m+1) \cup \ldots\\ &\ldots \cup \Theta(\delta_l, n, t_l[n])\eta(\delta, n, m+1)\theta(\delta, n, m+1)\end{aligned}$$

where $\theta(\delta, n, m+1)$ is a total unifier of $\rho(\delta, n, m+1)\eta(\delta, n, m+1)$.

Remark 1. The schematic unification defined above is based on local total unifiers which are unifiers over the schematic term language. We do not need higher-order substitutions as were used in [15].

Example 3. Let $(\mathcal{R}_b, \mathcal{R}_s)$ be the refutation schema defined in Example 2. The variables in $\rho(\delta^b, n, 0)$ are x, y, u, v, w. Therefore

$$\eta(\delta^b, n, 0) = \{x \leftarrow X_1^{\delta^b}(0), y \leftarrow X_2^{\delta^b}(0), u \leftarrow X_3^{\delta^b}(0), v \leftarrow X_4^{\delta^b}(0), w \leftarrow X_5^{\delta^b}(0)\}.$$

A total unifier of $\rho(\delta^b, n, 0)\eta(\delta^b, n, 0)$ is

$$\theta(\delta^b, n, 0)\colon \{X_2^{\delta^b}(0) \leftarrow X_1^{\delta^b}(0),\ X_4^{\delta^b}(0) \leftarrow X_3^{\delta^b}(0),\ X_5^{\delta^b}(0) \leftarrow X_3^{\delta^b}(0)\}$$

and $\Theta(\delta^b, n, 0) = \{\theta(\delta^b, n, 0)\}$. For $\rho(\delta, n, 0)$ we obtain

$$\theta(\delta, n, 0) = \{X_2^{\delta}(0) \leftarrow X_1^{\delta}(0),\ X_3^{\delta}(0) \leftarrow X_1^{\delta}(0)\}$$

and $\Theta(\delta, n, 0) = \{\theta(\delta, n, 0)\}$.

$$\theta(\delta, n, m+1) = \{X_1^{\delta}(m+1) \leftarrow \hat{f}(m, a), X_2^{\delta}(m+1) \leftarrow \hat{f}(m, a),\ X_3^{\delta}(m+1) \leftarrow \hat{f}(m, a)\}$$

and $\Theta(\delta, n, m+1) = \Theta(\delta, n, m) \cup \{\theta(\delta, n, m+1)\}$.
Finally we obtain

$$\theta(\delta^s, n, 0) = \{X_2^{\delta^s}(0) \leftarrow X_1^{\delta^s}(0)\}, \qquad \Theta(\delta^s, n, 0) = \{\theta(\delta^s, n, 0)\} \cup \Theta(\delta, n, n+1).$$

4 An application to a variant of the infinitary pigeonhole principle (IPP)

The resolution calculus for schematic clause sets defined in Section 3 is more powerful then the former schematic clause calculus defined in [15] and the superposition calculus used in [19] for schematic cut-elimination. In fact, we can provide an example refutation of a recursive clause set, which cannot be finitely represented in existing formalisms.

To this aim we will consider a formal proof of the infinitary pigeonhole principle whose cut-structure enforces a strict monotone order on the assignment of pigeons to holes, that is either each pigeon is assigned to a different hole by some linear ordering of the holes or once two pigeons are assigned to the same hole, all further pigeons are assigned to that hole. The cut-structure considered in this formal proof is even weaker than that of the eventually constant schema discussed in [11]. Mathematically this formal statement proves the infinitary pigeonhole principle when one assumes very weak properties of the assignment function.

We formalize this theory using a function $f(\cdot) : \iota \to \omega$. Our formalization uses two predicate symbols $=: \omega \to \omega \to o$ and $<: \omega \to \omega \to o$ whose properties are defined by the axioms of Definition 10. We also require the concept of successor and least element and thus add a monadic function $\mathbf{S}(\cdot) : \iota \to \iota$ and a constant $\mathbf{0} : \iota$ to the individual sort. We mark the individual sort successor and zero with bold to differentiate them from the numeric sort counterparts. Our formalization uses the following theory:

Definition 10 (order theory).

$$\left(\bigvee_{i=0}^{k} i = f(x)\right) \to f(x) < s(k) \tag{1}$$

$$\left(\bigvee_{i=0}^{k} i = f(\mathbf{s}(x))\right) \to f(x) < s(k) \tag{2}$$

$$(k = f(x) \land k = f(\mathbf{s}(x))) \to f(x) = f(\mathbf{s}(x)) \tag{3}$$

$$\neg f(\mathbf{0}) < 0 \tag{4}$$

$$f(\mathbf{s}(x)) < s(k) \to (k = f(\mathbf{s}(x)) \lor f(x) < k) \tag{5}$$

$$f(x) < s(k) \to (k = f(x) \lor f(x) < k) \tag{6}$$

Note that the first two statements of the theory contain iterations which is typically not allowed. One should interpret these iterations as abbreviation of the following schema of axioms:

$$0 = f(x) \vdash f(x) < s(k), \quad 1 = f(x) \vdash f(x) < s(k), \quad \cdots \quad k = f(x) \vdash f(x) < s(k)$$

Any axiom containing k is an axiom schema where k can range over the numeric sort. These statements can either be added to the context of the sequents or, by replacing the $\to$ by the sequent symbol, considered as an axiomatic extension of the **LK***-calculus.*

The end-sequent and the sequence of cut formulas are as follows:

Definition 11 (end-sequent IPP-schema variant).

$$\forall x \bigvee_{i=0}^{n} i = f(x) \vdash \exists x (f(x) = f(\mathbf{s}(x)))$$

where n ranges over the numeric sort.

Note that we do not include our order theory in the context of our end-sequent, rather we use the theory as initial sequents.

Definition 12 (sequence of cut formulas).

$$\exists x(n = f(x) \wedge n = f(\mathbf{S}(x))) \vee \forall x(f(x) < n)$$

where n ranges over the numeric sort.

Definition 13. *Let $\mathcal{C}^*(n)$ be the clause set resulting from the following schematic NNF definition Ψ:*

$$\begin{aligned}
Top(0) &= Next(0) \ \wedge \ (0 = f(\mathbf{0}) \vee 0 = f(\mathbf{S}(\mathbf{0}))) \\
Top(n+1) &= \forall x((n+1) = f(\mathbf{S}(x)) \vee f(x) < (n+1)) \ \wedge \\
&\quad\ \forall x((n+1) = f(x) \vee f(x) < (n+1)) \ \wedge Next(n+1) \\
Next(0) &= (\neg f(\mathbf{0}) < 0) \ \wedge \ \forall x((\neg 0 = f(x)) \vee (\neg 0 = f(\mathbf{S}(x)))) \\
Next(n+1) &= \forall x((\neg(n+1) = f(x)) \vee (\neg(n+1) = f(\mathbf{S}(x)))) \ \wedge \\
&\quad\ \forall x((\neg f(x) < (n+1)) \vee n = f(x) \vee f(x) < n) \ \wedge \\
&\quad\ \forall x((\neg f(\mathbf{S}(x)) < (n+1)) \vee n = f(\mathbf{S}(x)) \vee f(x) < n) \ \wedge Next(n)
\end{aligned}$$

Note that the clause set $\mathcal{C}^*(n)$ was extracted from a schematic proof by the schematic CERES method [19] and is therefore always unsatisfiable. Furthermore, rather than considering the clause set $\mathcal{C}^*(n)$ directly, we can instead consider the clause set resulting from the replacement of $\forall x((n+1) = f(\mathbf{S}(x)) \vee f(x) < (n+1))$ and $\forall x((n+1) = f(x) \vee f(x) < (n+1))$ by sequences of term instantiations for analysis. We will refer to this clause set as $\mathcal{C}(n)$. Notice that unsatisfiability of $\mathcal{C}(n)$ implies unsatisfiability of $\mathcal{C}^*(n)$. To define $\mathcal{C}(n)$ we need a few auxiliary definitions. First of all, we need to be able to translate between the individual sort and the numeric sort.

Definition 14 (translation between individual and numeric sort). *Let Sw be inductively defined as follows:*

$$Sw(0) \equiv \mathbf{0} \qquad\qquad Sw(n+1) \equiv \mathbf{S}(Sw(n))$$

Rather than directly refuting $\mathcal{C}^*(n)$ we consider the clause set $\mathcal{C}(n)$ which is derivable from $\mathcal{C}^*(n)$ by replacing the two clauses of $Top(n)$ by the following sequences of term instantiations.

Definition 15.

$$\begin{aligned}
Seq_1(0,n) &\equiv (n = f(\mathbf{S}(Sw(0))) \vee f(Sw(0)) < n) \\
Seq_1(m+1,n) &\equiv (n = f(\mathbf{S}(Sw(m+1))) \vee f(Sw(m+1)) < n) \ \wedge Seq_1(m,n) \\
Seq_2(0,n) &\equiv (n = f(Sw(0)) \vee f(Sw(0)) < n) \\
Seq_2(m+1,n) &\equiv (n = f(Sw(m+1)) \vee f(Sw(m+1)) < n) \ \wedge \ Seq_2(m,n)
\end{aligned}$$

Seq_1 and Seq_2 are used to define the clause set $\mathcal{C}(n) \equiv Seq_1(n,n) \wedge Seq_2(n,n) \wedge Next(n)$. The definitional clause sets of Seq_1 and Seq_2 are:

$$\begin{aligned}
\Phi(Seq_1(0,n), L_{Seq_1}(0,n)) = \{&L_{Seq_1}(0,n) \vdash L_{S_1}(0,n);\; L_{S_1}(0,n) \vdash L_{S_{11}}(0,n), L_{S_{12}}(0,n);\\
&L_{S_{11}}(0,n) \vdash n = f(\mathbf{S}(Sw(0)));\; L_{S_{12}}(0,n) \vdash f(Sw(0)) < n\}\\
\Phi(Seq_2(0,n), L_{Seq_2}(0,n)) = \{&L_{Seq_2}(0,n) \vdash L_{S_2}(0,n);\; L_{S_2}(0,n) \vdash L_{S_{21}}(0,n), L_{S_{22}}(0,n);\\
&L_{S_{21}}(0,n) \vdash n = f(Sw(0));\; L_{S_{22}}(0,n) \vdash f(Sw(0)) < n\}
\end{aligned}$$

$$\begin{aligned}
\Phi(Seq_1(m+1,n), L_{Seq_1}(m+1,n)) = \{&L_{Seq_1}(m+1,n) \vdash L_{S_1}(m+1,n);\\
&L_{Seq_1}(m+1,n) \vdash L_{Seq_1}(m,n);\\
&L_{S_1}(m+1,n) \vdash L_{S_{11}}(m+1,n), L_{S_{12}}(m+1,n);\\
&L_{S_{11}}(m+1,n) \vdash n = f(\mathbf{S}(Sw(m+1)));\\
&L_{S_{12}}(m+1,n) \vdash f(Sw(m+1)) < n\}\\
\Phi(Seq_2(m+1,n), L_{Seq_2}(m+1,n)) = \{&L_{Seq_2}(m+1,n) \vdash L_{S_2}(m+1,n);\\
&L_{Seq_2}(m+1,n) \vdash L_{Seq_2}(m,n);\\
&L_{S_2}(m+1,n) \vdash L_{S_{21}}(m+1,n), L_{S_{22}}(m+1,n);\\
&L_{S_{21}}(m+1,n) \vdash n = f(Sw(m+1));\\
&L_{S_{22}}(m+1,n) \vdash f(Sw(m+1)) < n\}
\end{aligned}$$

Lemma 1. *$\mathcal{C}(n)$ is derivable from $\mathcal{C}^*(n)$.*

Proof (sketch). Note that $\mathcal{C}(n) = Seq_1(Sw(n), n) \;\wedge\; Seq_2(Sw(n), n) \wedge Next(n)$ can be constructed from $\mathcal{C}^*(n)$ using the clauses

$$\forall x((\neg f(x) < (n+1)) \vee n = f(x) \vee f(x) < n)$$

$$\forall x((\neg f(\mathbf{S}(x)) < (n+1)) \vee n = f(\mathbf{S}(x)) \vee f(x) < n)$$

We perform our analysis on $\mathcal{C}(n)$ rather than $\mathcal{C}^*(n)$ because through $\mathcal{C}(n)$ we provide a necessary and sufficient condition for provability. In particular, Theorem 1 shows that the instantiations of $\mathcal{C}(n)$ which we have chosen are sufficient for refuting $\mathcal{C}^*(n)$, while Theorem 2 together with the fact that we are forced to use the atom $f(\mathbf{0}) < 0$ in any refutation provides a necessary condition for refuting the clause set.

Theorem 1. *For all $n > 1$, $\mathcal{C}(n) \vdash$ is provable in* LK *without cut from initial sequents of the form $A \vdash A$.*

By induction on n. Thus, by *branch* we are referring to the application $\vee : l$. If a branch contains both a negative and a positive instances of a literal (that is P and $\neg P$ then it may be *closed.* let $n = 1$, then

$$\mathcal{C}(1) \equiv Seq_1(\mathbf{S}(\mathbf{0}), 1) \;\wedge\; Seq_2(\mathbf{S}(\mathbf{0}), 1) \wedge Next(1)$$

and we must show that $\mathcal{C}(1) \vdash$ is provable. Unrolled $\mathcal{C}(1)$ results in the sequent $\Gamma \vdash$ which contains the following formulas:

$$(1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0}))) \vee f(\mathbf{S}(\mathbf{0})) < 1) \tag{7}$$
$$(1 = f(\mathbf{S}(\mathbf{0})) \vee f(\mathbf{0}) < 1) \tag{8}$$
$$(1 = f(\mathbf{S}(\mathbf{0})) \vee f(\mathbf{S}(\mathbf{0})) < 1) \tag{9}$$
$$(1 = f(\mathbf{0}) \vee f(\mathbf{0}) < 1) \tag{10}$$
$$\forall x((\neg 1 = f(x)) \vee (\neg 1 = f(\mathbf{S}(x)))) \tag{11}$$
$$\forall x((\neg f(x) < 1) \vee 0 = f(x) \vee f(x) < 0) \tag{12}$$
$$\forall x((\neg f(\mathbf{S}(x)) < 1) \vee 0 = f(\mathbf{S}(x)) \vee f(x) < 0) \tag{13}$$
$$(\neg f(\mathbf{0}) < 0) \tag{14}$$
$$\forall x((\neg 0 = f(x)) \vee (\neg 0 = f(\mathbf{S}(x)))) \tag{15}$$

Now let us consider the 16 sequents constructed from the clauses (7)−(10) where Δ contains clauses (11)−(15). These sequents are the leaves of the tree found in Figure 1.

$$1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0})))\ ,\ 1 = f(\mathbf{S}(\mathbf{0}))\ ,\ 1 = f(\mathbf{S}(\mathbf{0})))\ ,\ 1 = f(\mathbf{0}), \Delta \vdash \tag{16}$$
$$1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0})))\ ,\ f(\mathbf{0}) < 1\ ,\ 1 = f(\mathbf{S}(\mathbf{0})))\ ,\ 1 = f(\mathbf{0}), \Delta \vdash \tag{17}$$
$$f(\mathbf{S}(\mathbf{0})) < 1)\ ,\ 1 = f(\mathbf{S}(\mathbf{0}))\ ,\ 1 = f(\mathbf{S}(\mathbf{0})))\ ,\ 1 = f(\mathbf{0}), \Delta \vdash \tag{18}$$
$$f(\mathbf{S}(\mathbf{0})) < 1)\ ,\ f(\mathbf{0}) < 1\ ,\ 1 = f(\mathbf{S}(\mathbf{0})))\ ,\ 1 = f(\mathbf{0}), \Delta \vdash \tag{19}$$
$$1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0})))\ ,\ 1 = f(\mathbf{S}(\mathbf{0}))\ ,\ f(\mathbf{S}(\mathbf{0}))) < 1\ ,\ 1 = f(\mathbf{0}), \Delta \vdash \tag{20}$$
$$1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0})))\ ,\ f(\mathbf{0}) < 1\ ,\ f(\mathbf{S}(\mathbf{0}))) < 1\ ,\ 1 = f(\mathbf{0}), \Delta \vdash \tag{21}$$
$$f(\mathbf{S}(\mathbf{0})) < 1)\ ,\ 1 = f(\mathbf{S}(\mathbf{0}))\ ,\ f(\mathbf{S}(\mathbf{0}))) < 1\ ,\ 1 = f(\mathbf{0}), \Delta \vdash \tag{22}$$
$$f(\mathbf{S}(\mathbf{0})) < 1)\ ,\ f(\mathbf{0}) < 1\ ,\ f(\mathbf{S}(\mathbf{0}))) < 1\ ,\ 1 = f(\mathbf{0}), \Delta \vdash \tag{23}$$
$$1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0})))\ ,\ 1 = f(\mathbf{S}(\mathbf{0}))\ ,\ 1 = f(\mathbf{S}(\mathbf{0})))\ ,\ f(\mathbf{0}) < 1, \Delta \vdash \tag{24}$$
$$1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0})))\ ,\ f(\mathbf{0}) < 1\ ,\ 1 = f(\mathbf{S}(\mathbf{0})))\ ,\ f(\mathbf{0}) < 1, \Delta \vdash \tag{25}$$
$$f(\mathbf{S}(\mathbf{0})) < 1)\ ,\ 1 = f(\mathbf{S}(\mathbf{0}))\ ,\ 1 = f(\mathbf{S}(\mathbf{0})))\ ,\ f(\mathbf{0}) < 1, \Delta \vdash \tag{26}$$
$$f(\mathbf{S}(\mathbf{0})) < 1)\ ,\ f(\mathbf{0}) < 1\ ,\ 1 = f(\mathbf{S}(\mathbf{0})))\ ,\ f(\mathbf{0}) < 1, \Delta \vdash \tag{27}$$
$$1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0})))\ ,\ 1 = f(\mathbf{S}(\mathbf{0}))\ ,\ f(\mathbf{S}(\mathbf{0}))) < 1\ ,\ f(\mathbf{0}) < 1, \Delta \vdash \tag{28}$$
$$1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0})))\ ,\ f(\mathbf{0}) < 1\ ,\ f(\mathbf{S}(\mathbf{0}))) < 1\ ,\ f(\mathbf{0}) < 1, \Delta \vdash \tag{29}$$
$$f(\mathbf{S}(\mathbf{0})) < 1)\ ,\ 1 = f(\mathbf{S}(\mathbf{0}))\ ,\ f(\mathbf{S}(\mathbf{0}))) < 1\ ,\ f(\mathbf{0}) < 1, \Delta \vdash \tag{30}$$
$$f(\mathbf{S}(\mathbf{0})) < 1)\ ,\ f(\mathbf{0}) < 1\ ,\ f(\mathbf{S}(\mathbf{0}))) < 1\ ,\ f(\mathbf{0}) < 1, \Delta \vdash \tag{31}$$

If a branch contains both $1 = f(\mathbf{S}(\mathbf{0}))$ and $1 = f(\mathbf{0})$ or $1 = f(\mathbf{S}(\mathbf{0}))$ and $1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0})))$ then it can be closed using (11) from Δ. For example consider branch (16)

$$\dfrac{\dfrac{\dfrac{\dfrac{16 \quad 17}{(10)}\vee:l \quad \dfrac{18 \quad 19}{(10)}\vee:l}{(9)}\vee:l \quad \dfrac{\dfrac{20 \quad 21}{(10)}\vee:l \quad \dfrac{22 \quad 23}{(10)}\vee:l}{(9)}\vee:l}{(8)}\vee:l \quad \dfrac{\dfrac{\dfrac{24 \quad 25}{(10)}\vee:l \quad \dfrac{26 \quad 27}{(10)}\vee:l}{(9)}\vee:l \quad \dfrac{\dfrac{28 \quad 29}{(10)}\vee:l \quad \dfrac{30 \quad 31}{(10)}\vee:l}{(9)}\vee:l}{(8)}\vee:l}{(7)}\vee:l$$

Fig. 1. Proof tree construction.

$$\dfrac{\dfrac{\dfrac{\dfrac{1 = f(\mathbf{S}(\mathbf{0}))) \vdash 1 = f(\mathbf{S}(\mathbf{0}))}{1 = f(\mathbf{S}(\mathbf{0}))), \neg 1 = f(\mathbf{S}(\mathbf{0})), \Delta \vdash} \quad \dfrac{1 = f(\mathbf{0}) \vdash 1 = f(\mathbf{0})}{1 = f(\mathbf{0}), \neg 1 = f(\mathbf{0})), \Delta \vdash}}{1 = f(\mathbf{S}(\mathbf{0}))) \ , \ 1 = f(\mathbf{0}), (\neg 1 = f(\mathbf{0})) \vee (\neg 1 = f(\mathbf{S}(\mathbf{0}))), \Delta \vdash}}{1 = f(\mathbf{S}(\mathbf{0}))) \ , \ 1 = f(\mathbf{0}), \forall x((\neg 1 = f(x)) \vee (\neg 1 = f(\mathbf{S}(x)))), \Delta \vdash}}{1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0}))) \ , \ 1 = f(\mathbf{S}(\mathbf{0})) \ , \ 1 = f(\mathbf{S}(\mathbf{0}))) \ , \ 1 = f(\mathbf{0}), \Delta \vdash}$$

This leaves the following branches:

$$1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0}))) \ , \ f(\mathbf{0}) < 1 \ , \ f(\mathbf{S}(\mathbf{0}))) < 1 \ , \ 1 = f(\mathbf{0}), \Delta \vdash \tag{32}$$
$$f(\mathbf{S}(\mathbf{0})) < 1) \ , \ f(\mathbf{0}) < 1 \ , \ f(\mathbf{S}(\mathbf{0}))) < 1 \ , \ 1 = f(\mathbf{0}), \Delta \vdash \tag{33}$$
$$f(\mathbf{S}(\mathbf{0})) < 1) \ , \ 1 = f(\mathbf{S}(\mathbf{0})) \ , \ 1 = f(\mathbf{S}(\mathbf{0}))) \ , \ f(\mathbf{0}) < 1, \Delta \vdash \tag{34}$$
$$f(\mathbf{S}(\mathbf{0})) < 1) \ , \ f(\mathbf{0}) < 1 \ , \ 1 = f(\mathbf{S}(\mathbf{0}))) \ , \ f(\mathbf{0}) < 1, \Delta \vdash \tag{35}$$
$$f(\mathbf{S}(\mathbf{0})) < 1) \ , \ 1 = f(\mathbf{S}(\mathbf{0})) \ , \ f(\mathbf{S}(\mathbf{0}))) < 1 \ , \ f(\mathbf{0}) < 1, \Delta \vdash \tag{36}$$
$$f(\mathbf{S}(\mathbf{0})) < 1) \ , \ f(\mathbf{0}) < 1 \ , \ f(\mathbf{S}(\mathbf{0}))) < 1 \ , \ f(\mathbf{0}) < 1, \Delta \vdash \tag{37}$$

We can simplify these sequents by application of weakening and contraction rules so that they are all of the form

$$f(\mathbf{0}) < 1 \ , \ f(\mathbf{S}(\mathbf{0}))) < 1, \Delta \vdash \tag{38}$$

Using these two literals and (12) & (13) we derive a sequent containing the following three clauses:

$$0 = f(\mathbf{0}) \vee f(\mathbf{0}) < 0 \tag{39}$$
$$0 = f(\mathbf{S}(\mathbf{0})) \vee f(\mathbf{S}(\mathbf{0})) < 0 \tag{40}$$
$$0 = f(\mathbf{S}(\mathbf{0})) \vee f(\mathbf{0}) < 0 \tag{41}$$

We will refer to the sequent containing (39), (40), and (41) as $\mathcal{S}$ and can construct the following proof tree starting at $\mathcal{S}$

$$\dfrac{\dfrac{\dfrac{42 \quad 43}{(41)}\vee : l \quad \dfrac{44 \quad 45}{(41)}\vee : l}{(40)}\vee : l \quad \dfrac{\dfrac{46 \quad 47}{(41)}\vee : l \quad \dfrac{48 \quad 49}{(41)}\vee : l}{(40)}\vee : l}{(39)}\vee : l$$

Where the leaves are the following sequents

$$0 = f(\mathbf{0}), 0 = f(\mathbf{S}(\mathbf{0})), 0 = f(\mathbf{S}(\mathbf{0})), \Delta \vdash \tag{42}$$
$$0 = f(\mathbf{0}), f(\mathbf{S}(\mathbf{0})) < 0, 0 = f(\mathbf{S}(\mathbf{0})), \Delta \vdash \tag{43}$$
$$0 = f(\mathbf{0}), 0 = f(\mathbf{S}(\mathbf{0})), f(\mathbf{0}) < 0, \Delta \vdash \tag{44}$$
$$0 = f(\mathbf{0}), f(\mathbf{S}(\mathbf{0})) < 0, f(\mathbf{0}) < 0, \Delta \vdash \tag{45}$$
$$f(\mathbf{0}) < 0, 0 = f(\mathbf{S}(\mathbf{0})), 0 = f(\mathbf{S}(\mathbf{0})), \Delta \vdash \tag{46}$$
$$f(\mathbf{0}) < 0, f(\mathbf{S}(\mathbf{0})) < 0, 0 = f(\mathbf{S}(\mathbf{0})), \Delta \vdash \tag{47}$$
$$f(\mathbf{0}) < 0, 0 = f(\mathbf{S}(\mathbf{0})) f(\mathbf{0}) < 0, \Delta \vdash \tag{48}$$
$$f(\mathbf{0}) < 0, f(\mathbf{S}(\mathbf{0})) < 0, f(\mathbf{0}) < 0, \Delta \vdash \tag{49}$$

Sequent (44)−(49) contain $f(\mathbf{0}) < 0$, and thus can be closed using (14). This leaves sequents (42) and (43) which can be closed using (15). Thus we have shown that $\mathcal{C}(1) \vdash$ is provable. For the step case we assume that $\mathcal{C}(n) \vdash$ is provable and show that $\mathcal{C}(n+1) \vdash$ is provable.

Let us consider the sequent $Seq_1(Sw(n), n), Seq_2(Sw(n), n), Next(n) \vdash$ which can be constructed from $\mathcal{C}^*(n+1) \vdash$ using the clauses

$$\forall x((\neg f(x) < (n+1)) \vee n = f(x) \vee f(x) < n) \tag{50}$$
$$\forall x((\neg f(\mathbf{S}(x)) < (n+1)) \vee n = f(\mathbf{S}(x)) \vee f(x) < n) \tag{51}$$

found in $Next(n+1)$. We denote the following literal set by Γ:

$$f(\mathbf{0}) < (n+1)\ ,\ f(\mathbf{S}(\mathbf{0}))) < (n+1)\ ,\ \cdots\ ,\ f(\mathbf{S}(Sw(n))) < (n+1) \tag{52}$$

These observations entail the soundness of the following proof tree:

$$\dfrac{\dfrac{Seq_1(Sw(n), n), Seq_2(Sw(n), n), Next(n) \vdash}{\vdots}}{\Gamma, Next(n+1) \vdash}$$

Notice the similarities between $\Gamma, Next(n+1) \vdash$ and (38). Essentially, if we consider the sequent $Seq_1(Sw(n+1), n+1), Seq_2(Sw(n+1), n+1), Next(n+1) \vdash$, apply $\vee : l$ to each clause of $Seq_1(Sw(n+1), n+1)$ and $Seq_2(Sw(n+1), n+1)$ one, close all branches where $\forall x((\neg(n+1) = f(x)) \vee (\neg(n+1) = f(\mathbf{S}(x))))$ can be applied, and finally apply weakenings and contraction as in the basecase $\Gamma, Next(n+1) \vdash$ would be the resulting sequent. To see that this would be the case we would have to show that the sequents at the leaves of this construction can either be closed by $\forall x((\neg(n+1) = f(x)) \vee (\neg(n+1) = f(\mathbf{S}(x))))$ or they contain Γ. This follows directly from the definition of Seq_1 and Seq_2. Notice that $Seq_1(k, n+1)$ and $Seq_2(k, n+1)$ for some $0 \leq k \leq n+1$ add the following two formula to the sequent:

$$(n+1) = f(\mathbf{S}(k)) \vee f(k) < (n+1) \tag{53}$$
$$(n+1) = f(k) \vee f(k) < (n+1) \tag{54}$$

From these two formula we can make the following four sequents:

$$(n+1) = f(\mathbf{S}(k)), (n+1) = f(k), \Delta \vdash \tag{55}$$
$$(n+1) = f(\mathbf{S}(k)), f(k) < (n+1), \Delta \vdash \tag{56}$$
$$f(k) < (n+1), f(k) < (n+1), \Delta \vdash \tag{57}$$
$$f(k) < (n+1), (n+1) = f(k), \Delta \vdash \tag{58}$$

Notice that only (55) can be closed by $\forall x((\neg(n+1) = f(x)) \vee (\neg(n+1) = f(\mathbf{S}(x))))$, but every other branch contains at least one instance of $(n+1) = f(k)$. Thus, every branch which is not closed will contain at least one instance $(n+1) = f(k)$ for each $0 \leq k \leq n+1$. This is precisely Γ. Thus it follows that the proof tree

$$\cfrac{\cfrac{\cfrac{Seq_1(Sw(n),n), Seq_2(Sw(n),n), Next(n) \vdash}{\vdots}}{\Gamma, Next(n+1) \vdash}}{\cfrac{\vdots}{Seq_1(Sw(n+1),n+1), Seq_2(Sw(n+1),n+1), Next(n+1) \vdash}}$$

is a sound derivation. The root of this derivation is precisely $\mathcal{C}(n{+}1) \vdash$. By the induction hypothesis that $\mathcal{C}(n) \vdash$ is provable which is the leave sequent of every open branch we see that $\mathcal{C}(n{+}1) \vdash$ is also provable. Note that we are considering provability of a negative statement in LK without cut rather than provability by resolution. Essentially, these questions are the same. Furthermore we only consider provability of instances rather than provability of the entire sequence at once. This is justified by the fact that sequence provability puts additional restrictions on the proof construction, for example finite representability. This is also the reason for the provability of the very weak pigeonhole principle to be a non-trivial task, one needs n instances of a clause indexed by n which implies that a non-finite number of inferences needs to be performed in a finite number of steps.

Theorem 1 shows that the instantiations of $\mathcal{C}(n)$ are sufficient for showing the unsatisfiability of $\mathcal{C}^*(n)$. To show that these instantiations are also necessary let us consider a restriction of $C(n)$ which only contains $2n-1$ of the $2n$ instances derived from $\mathcal{C}^*(n)$. This results in a sequence of clause sets we refer to as $C_1(n,k)$ and $C_2(n,k)$, where $0 \leq k \leq n$, which use the following variants of Seq_1 and Seq_2.

Definition 16.

$$
\begin{aligned}
Seq_1'(k+1,n,k+1) &\equiv Seq_1'(k,n,k+1)\\
Seq_1'(m+1,n,k) &\equiv (n = f(\mathbf{S}(sw(m+1))) \vee f(sw(m+1)) < n) \ \wedge \ Seq_1'(m,n,k)\\
Seq_1'(0,n,0) &\equiv \top\\
Seq_1'(0,n,k) &\equiv (n = f(\mathbf{S}(sw(0))) \vee f(sw(0)) < n)\\
Seq_2'(k+1,n,k+1) &\equiv Seq_2'(k,n,k+1)\\
Seq_2'(m+1,n,k) &\equiv (n = f(sw(m+1)) \vee f(sw(m+1)) < n) \ \wedge \ Seq_2'(m,n,k)\\
Seq_2'(0,n,0) &\equiv \top\\
Seq_2'(0,n,k) &\equiv (n = f(sw(0)) \vee f(sw(0)) < n)
\end{aligned}
$$

We define $C_1(n,k) = Seq_1'(k, Sw(n), n) \ \wedge \ Seq_2(Sw(n), n) \wedge Next(n)$ and $C_2(n,k) = Seq_1(Sw(n), n) \ \wedge \ Seq_2'(k, Sw(n), n) \wedge Next(n)$. Notice that for $Seq_1'(m,n,k)$ and $Seq_2'(m,n,k)$ we skip the instantiation when $m = k$.

Theorem 2. *For all $n > 1$ and $0 \leq k \leq n$, $C_1(n,k) \vdash$ and $C_2(n,k) \vdash$ are not provable in* LK *without cut from initial sequents of the form $A \vdash A$.*

Proof. By induction on n and by case distinction on k. let $n = 1$, then we are considering the following four sequents $\mathcal{C}_1(1,0) \vdash$, $\mathcal{C}_2(1,0) \vdash$, $\mathcal{C}_1(1,1) \vdash$, $\mathcal{C}_2(1,1) \vdash$. When Δ denotes clauses (11)$-$(15), we can represent the sequents as follows:

$$seq_1'(Sw(1),1,0) \ , \ Seq_2(Sw(1),1), \Delta \vdash \tag{59}$$
$$seq_1'(Sw(1),1,1) \ , \ Seq_2(Sw(1),1), \Delta \vdash \tag{60}$$
$$Seq_1(Sw(1),1) \ , \ Seq_2'(Sw(1),1,0), \Delta \vdash \tag{61}$$
$$Seq_1(Sw(1),1) \ , \ Seq_2'(Sw(1),1,1), \Delta \vdash \tag{62}$$

We can replace seq_1 ,seq_1', seq_2, and seq_2' by formula as follows:

(59)

$$(1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0}))) \vee f(\mathbf{S}(\mathbf{0})) < 1) \tag{63}$$
$$(1 = f(\mathbf{S}(\mathbf{0})) \vee f(\mathbf{S}(\mathbf{0})) < 1) \tag{64}$$
$$(1 = f(\mathbf{0}) \vee f(\mathbf{0}) < 1) \tag{65}$$

(60)

$$(1 = f(\mathbf{S}(\mathbf{0})) \vee f(\mathbf{0}) < 1) \tag{66}$$
$$(1 = f(\mathbf{S}(\mathbf{0})) \vee f(\mathbf{S}(\mathbf{0})) < 1) \tag{67}$$
$$(1 = f(\mathbf{0}) \vee f(\mathbf{0}) < 1) \tag{68}$$

(61)

$$(1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0}))) \vee f(\mathbf{S}(\mathbf{0})) < 1) \tag{69}$$
$$(1 = f(\mathbf{S}(\mathbf{0})) \vee f(\mathbf{0}) < 1) \tag{70}$$
$$(1 = f(\mathbf{S}(\mathbf{0})) \vee f(\mathbf{S}(\mathbf{0})) < 1) \tag{71}$$

(62)

$$(1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0}))) \vee f(\mathbf{S}(\mathbf{0})) < 1) \tag{72}$$
$$(1 = f(\mathbf{S}(\mathbf{0})) \vee f(\mathbf{0}) < 1) \tag{73}$$
$$(1 = f(\mathbf{0}) \vee f(\mathbf{0}) < 1) \tag{74}$$

As was done in the case of Theorem **??** we can apply $\vee : l$ to each of the formula derived above and consider which branches can be closed by formula of Δ and which cannot be. In this case Δ is $next(1)$. Below we highlight the branches derived from the above sequents which cannot be closed using (11).

(59)

$$f(\mathbf{S}(\mathbf{0})) < 1\ ,\ f(\mathbf{S}(\mathbf{0})) < 1\ ,\ 1 = f(\mathbf{0}), \Delta \vdash \tag{75}$$

(60)

$$f(\mathbf{0}) < 1\ ,\ 1 = f(\mathbf{S}(\mathbf{0}))\ ,\ f(\mathbf{0}) < 1, \Delta \vdash \tag{76}$$

(61)

$$f(\mathbf{S}(\mathbf{0})) < 1\ ,\ 1 = f(\mathbf{S}(\mathbf{0}))\ ,\ f(\mathbf{S}(\mathbf{0})) < 1, \Delta \vdash \tag{77}$$

(62)

$$1 = f(\mathbf{S}(\mathbf{S}(\mathbf{0})))\ ,\ f(\mathbf{0}) < 1\ ,\ f(\mathbf{0}) < 1, \Delta \vdash \tag{78}$$

unlike the basecase in of Theorem 1 applying weakening and contraction results in more than one branch. In particular, this results in the following:

(60) & (62)

$$f(\mathbf{0}) < 1, \Delta \vdash \tag{79}$$

(59) & (61)

$$f(\mathbf{S}(\mathbf{0})) < 1, \Delta \vdash \tag{80}$$

Now let us consider the result of apply (12) and (13) to (79) and (80). The result is as follows

$$(60) \ \& \ (62)$$

$$0 = f(\mathbf{0}) \vee f(\mathbf{0}) < 0, \Delta \vdash \tag{81}$$

$$(59) \ \& \ (61)$$

$$(0 = f(\mathbf{S}(\mathbf{0})) \vee f(\mathbf{0}) < 0), (0 = f(\mathbf{S}(\mathbf{0})) \vee f(\mathbf{S}(\mathbf{0})) < 0), \Delta \vdash \tag{82}$$

The branch $0 = f(\mathbf{0}), \Delta \vdash$ entails from (81) and cannot be closed. Also the branch $0 = f(\mathbf{S}(\mathbf{0})), f(\mathbf{S}(\mathbf{0})) < 0, \Delta \vdash$ and cannot be closed. Thus, none of the sequents $\mathcal{C}_1(1,0) \vdash$, $\mathcal{C}_2(1,0) \vdash$, $\mathcal{C}_1(1,1) \vdash$, $\mathcal{C}_2(1,1) \vdash$ are provable.

Now let use assume that the sequents $\mathcal{C}_1(n,k) \vdash$ and $\mathcal{C}_2(n,k) \vdash$ are not provable for each $0 \leq k \leq n$, we now show that the clause sets $\mathcal{C}_1(n+1,k) \vdash$ and $\mathcal{C}_2(n+1,k) \vdash$ are not provable for any $0 \leq k \leq n+1$.

We prove the step case by a case distinction on k. when $k = n+1$, the missing clause from $\mathcal{C}_1(n+1,n+1)$ is

$$n + 1 = f(\mathbf{S}(sw(n+1))) \vee f(sw(n+1)) < n + 1 \tag{83}$$

and the missing clause of $\mathcal{C}_2(n+1,n+1)$ is

$$n + 1 = f(sw(n+1)) \vee f(sw(n+1)) < n + 1 \tag{84}$$

In the case of $\mathcal{C}_1(n+1,n+1)$ the only clause containing $f(sw(n+1)) < n+1$ is (84) and In the case of $\mathcal{C}_2(n+1,n+1)$ the only clause containing $f(sw(n+1)) < n+1$ is (83). Now consider the branches which have the following form:

$$f(\mathbf{0}) < n+1 \ , \ \cdots \ , \ f(sw(n)) < n+1 \ , \ n+1 = f(sw(n+1)) \tag{85}$$

$$f(\mathbf{0}) < n+1 \ , \ \cdots \ , \ f(sw(n)) < n+1 \ , \ n+1 = f(\mathbf{S}(sw(n+1))) \tag{86}$$

Being that neither branch contains the literal $f(sw(n+1)) < n+1$ we cannot derive the following formula:

$$n = f(sw(n+1)) \vee f(sw(n+1)) < n \tag{87}$$

$$n = f(\mathbf{S}(sw(n))) \vee f(sw(n)) < n \tag{88}$$

While (87) isn't really of interest $n = f(\mathbf{S}(sw(n))) \vee f(sw(n)) < n$ is a clause of $\mathcal{C}(n)$ which cannot be derived from

$$f(\mathbf{0}) < n+1 \ , \ \cdots \ , \ f(sw(n)) < n+1$$

and thus on this branch we can only derive $\mathcal{C}_1(n,n) \vdash$ and by the induction hypothesis $\mathcal{C}_1(n,n) \vdash$ is not provable. Now what is left to show is that the same can be done of $k < n+1$. The missing clause from $\mathcal{C}_1(n+1,k)$ is

$$n + 1 = f(\mathbf{S}(sw(k))) \vee f(sw(k)) < n + 1 \tag{89}$$

and the missing clause of $\mathcal{C}_2(n + 1, k)$ is

$$n + 1 = f(sw(k)) \vee f(sw(k)) < n + 1 \tag{90}$$

Like in the previous case, for $\mathcal{C}_1(n+1, k)$ the only clause containing $f(sw(k)) < n+1$ is (90) and in the case of $\mathcal{C}_2(n+1, k)$ the only clause containing $f(sw(k)) < n + 1$ is (89). Now consider the branches which have the following form:

$$f(\mathbf{0}) < n + 1 \ , \ \cdots \ , \ n + 1 = f(sw(k)) \ , \ \cdots \ , \ f(sw(n + 1)) < n + 1 \tag{91}$$
$$f(\mathbf{0}) < n + 1 \ , \ \cdots \ , \ n + 1 = f(\mathbf{S}(sw(k))) \ , \ \cdots \ , \ f(sw(n + 1)) < n + 1 \tag{92}$$

Being that neither branch contains the literal $f(sw(k)) < n + 1$ we cannot derive the following clauses:

$$n = f(sw(k)) \vee f(sw(k)) < n \tag{93}$$
$$n = f(\mathbf{S}(sw(k - 1))) \vee f(sw(k - 1)) < n \tag{94}$$

If $k = 0$ we only need to consider line (93) in which case we can only derive $\mathcal{C}_2(n, 0)$ on the branch which is satisfiable by the induction hypothesis. If $0 < k < n + 1$ then we can only derive the sequent

$$\mathit{Seq}'_1(\mathit{Sw}(n), n, k) \ \wedge \ \mathit{Seq}'_2(\mathit{Sw}(n), n, k) \wedge \mathit{Next}(n) \vdash$$

which is missing two clauses instead of just one and being that both $\mathcal{C}_1(n, k)$ and $\mathcal{C}_2(n, k)$ are not provable a subset of either one of them is also not provable. Thus we have proven the theorem by induction.

Theorem 2 implies that the formalism defined in [15] cannot be used to formalize a refutation of the clause set constructed from $\mathcal{C}^*(n)$, because n instances of a clause indexed by n are needed. In other words, a nested loop or a stack is needed to express the refutation finitely. Furthermore, $\mathcal{C}^*(\alpha) \vdash$, where α is a numeral, is a Σ_1 statement and hence, any proof of the sequent $\mathcal{C}^*(\alpha) \vdash$ requires Σ_1 induction. However, Σ_1-incompleteness of the superposition calculus used by the formalism in [19] has been conjectured in [24]. Therefore, the example above is most likely out of the scope of all existing formalisms. It is however possible to give a recursive refutation of $\mathcal{C}(n)$ in our calculus.

Theorem 3. *There exists a schematic resolution refutation of* $\mathrm{CL}(\Psi)$ *(Ψ as in Definition 13).*

Proof (sketch). Let us consider a variation of the schematic NNF Ψ where $\mathit{Top}(n + 1)$ is defined as follows:

$$\mathit{Top}(n + 1) \ \equiv \ \mathit{Seq}_1(n + 1, n + 1) \ \wedge \mathit{Seq}_2(n + 1, n + 1) \ \wedge \mathit{Next}(n + 1),$$

where $Seq_1(n,n)$ and $Seq_2(n,n)$ are constructed as in Definition 15. We refer to this schematic NNF as Φ (which in fact is the NNF of $\mathcal{C}(n)$). Notice that instantiations of Φ by a numeral n results in the clause set $\mathcal{C}(n)$ and thus the schematic NNFs Ψ and Φ are related in the same way $\mathcal{C}^*(n)$ and $\mathcal{C}(n)$ are related. Our choice to work with Φ rather than Ψ is merely syntactic sugar to make our presentation clearer. Otherwise we would have an additional recursion construction of the instances Φ which obfuscates the essential part of the refutation. A full construction using analytic cuts can be found in the example directory of the GAPT system[5] in the file *VeryWeakPHPRefutation.scala*. Now let us consider $\mathrm{CL}(\Phi)$ which contains the following definitional clauses:

$$\begin{aligned}\mathrm{CL}(\Phi,0) = \{&\vdash L_{Top}(0,x);\ L_{Top}(0,x) \vdash L_1(0,x);\ L_{Top}(0,x) \vdash L_{Next}(0,x);\\ &L_1(0,x) \vdash L_{11}(0,x), L_{12}(0,x);\ L_{11}(0,x) \vdash 0 = f(\mathbf{0});\ L_{12}(0,x) \vdash 0 = f(\mathbf{S}(\mathbf{0}));\\ &L_{Next}(0,x) \vdash L_3(0,x);\ L_{Next}(0,x) \vdash L_4(0,x);\ L_3(0,x) \vdash L_{31}(0,x), L_{32}(0,x);\\ &L_{31}(0,x), 0 = f(x) \vdash;\ L_{32}(0,x), 0 = f(\mathbf{S}(x))) \vdash;\ L_4(0,x), f(\mathbf{0}) < 0 \vdash\}\end{aligned}$$

[5] https://www.logic.at/gapt/ [16]

$$
\begin{aligned}
\mathrm{CL}(\Phi, n+1) = \{ & \vdash L_{Top}(n+1,x);\ L_{Top}(n+1,x) \vdash L_S(n+1,x);\\
& L_{Top}(n+1,x) \vdash L_{Next}(n+1,x);\\
& L_S(n+1,n+1) \vdash L_{Seq_1}(n+1,n+1);\\
& L_S(n+1,n+1) \vdash L_{Seq_2}(n+1,n+1);\\
& L_{Next}(n+1,x) \vdash L_1(n+1,x);\ L_{Next}(n+1,x) \vdash L_2(n+1,x);\\
& L_1(n+1,x) \vdash L_3(n+1,x);\ L_1(n+1,x) \vdash L_4(n+1,x);\\
& L_2(n+1,x) \vdash L_5(n+1,x);\ L_2(n+1,x) \vdash L_{Next}(n,x);\\
& L_3(n+1,x) \vdash L_6(n+1,x), L_7(n+1,x);\\
& L_6(n+1,x), (n+1) = f(x) \vdash;\ L_7(n+1,x), (n+1) = f(\mathbf{S}(x)) \vdash;\\
& L_4(n+1,x) \vdash L_9(n+1,x), L_{10}(n+1,x);\\
& L_9(n+1,x) \vdash L_{11}(n+1,x), L_{12}(n+1,x);\ L_{11}(n+1,x), f(x) < (n+1) \vdash;\\
& L_{12}(n+1,x) \vdash n = f(x);\ L_{10}(n+1,x) \vdash f(x) < n;\\
& L_5(n+1,x) \vdash L_{13}(n+1,x), L_{14}(n+1,x);\\
& L_{13}(n+1,x) \vdash L_{15}(n+1,x), L_{16}(n+1,x);\\
& L_{15}(n+1,x), f(\mathbf{S}(x)) < (n+1) \vdash;\ L_{16}(n+1,x) \vdash n = f(\mathbf{S}(x));\\
& L_{14}(n+1,x) \vdash f(x) < n; L_{Seq_1}(n+1,n+1) \vdash L_{S_1}(n+1,n+1);\\
& L_{Seq_1}(n+1,n+1) \vdash L_{Seq_1}(n,n+1);\\
& L_{S_1}(n+1,n+1) \vdash L_{S_{11}}(n+1,n+1), L_{S_{12}}(n+1,n+1);\\
& L_{S_{11}}(n+1,n+1) \vdash n+1 = f(\mathbf{S}(Sw(n+1)));\\
& L_{S_{12}}(n+1,n+1) \vdash f(Sw(n+1)) < n+1;\\
& L_{Seq_2}(n+1,n+1) \vdash L_{S_2}(n+1,n+1);\\
& L_{Seq_2}(n+1,n+1) \vdash L_{Seq_2}(n+1,n+1);\\
& L_{S_2}(n+1,n+1) \vdash L_{S_{21}}(n+1,n+1), L_{S_{22}}(n+1,n+1);\\
& L_{S_{21}}(n+1,n+1) \vdash n+1 = f(Sw(n+1));\\
& L_{S_{22}}(n+1,n+1) \vdash f(Sw(n+1)) < n+1\}
\end{aligned}
$$

Note that for simplicity reasons we are not using the full set $\mathrm{CL}(\Phi, n+1)$, but rather a subset and simplified version $\mathrm{CL}'(\Phi, n+1)$ (which can be obtained by renaming), that can easily be constructed from $\mathrm{CL}(\Phi, n+1)$.

$$
\mathrm{CL}'(\Phi, n{+}1) = \left\{ \begin{array}{ll}
(A): & \vdash L_{Top}(n+1,x) \\
(B): & L_{Top}(n+1,x) \vdash L_{Seq_1}(n+1,n+1) \\
(C): & L_{Top}(n+1,x) \vdash L_{Seq_2}(n+1,n+1) \\
(D): & L_{Top}(n+1,x) \vdash L_{Next}(n+1,x) \\
(E): & L_{Next}(n+1,x), (n+1) = f(\mathbf{S}(x)), (n+1) = f(x) \vdash \\
(F): & L_{Next}(n+1,x), f(x) < (n+1) \vdash n = f(x), f(x) < n \\
(G): & L_{Next}(n+1,x), f(\mathbf{S}(x)) < (n+1) \vdash n = f(\mathbf{S}(x)), f(x) < n \\
(H): & \mathbf{L_{Next}(n+1,x) \vdash L_{Next}(n,x)} \\
(I): & L_{Seq_2}(n+1,n+1) \vdash n+1 = f(Sw(n+1)), f(Sw(n+1)) < n+1 \\
(J): & L_{Seq_2}(n+1,n+1) \vdash L_{Seq_2}(n,n+1) \\
(K): & L_{Seq_1}(n+1,n+1) \vdash n+1 = f(\mathbf{S}(Sw(n+1))), f(Sw(n+1)) < n+1 \\
(L): & L_{Seq_1}(n+1,n+1) \vdash L_{Seq_1}(n,n+1)
\end{array} \right\}
$$

Clause (H) can be further transformed definitionally and thus represents the recursive call of the stepcase.

$$
\dfrac{(\mathrm{H}) \qquad \dfrac{\dfrac{\overset{(3)}{\Delta(n) \vdash \quad \Delta(n+1) \vdash L_{Seq_2}(n,n)}}{\Delta(n+1), L_{Next}(n,x), L_{Seq_1}(n,n) \vdash} \qquad \overset{(2)}{\Delta(n+1) \vdash L_{Seq_1}(n,n)}}{\Delta(n+1), L_{Next}(n,x) \vdash}}{\underset{(1)}{\Delta(n+1) \vdash}}
$$

$$
\dfrac{\dfrac{(\mathrm{B}) \quad (\mathrm{A})}{\vdash L_{Seq_1}(n+1,n+1)} \qquad \dfrac{\dfrac{(1) \quad \dfrac{(\mathrm{A}) \quad (\mathrm{D})}{\vdash L_{Next}(n+1,x)}}{L_{Seq_1}(n+1,n+1), L_{Seq_2}(n+1,n+1) \vdash} \qquad \dfrac{(\mathrm{C}) \quad (\mathrm{A})}{\vdash L_{Seq_2}(n+1,n+1)}}{L_{Seq_1}(n+1,n+1) \vdash}}{\vdash}
$$

where $\Delta(n) = L_{Seq_1}(n,n), L_{Seq_2}(n,n), L_{Next}(n,x)$. The loop labeled by (1) is the main loop of the resolution refutation. There are two more loops which start above labels (2) and (3) which construct the two sequences of quantifier instantiations. We do not go into further detail concerning the construction of these sub-refutations being that they are tedious and complex. However, what is interesting to note is that these loops are invariant on the following two clauses:

$$
L_{Seq_1}(k,n+1), L_{S_1}(k,n+1), L_{Seq_2}(k,n+1), L_{Next}(n+1,x) \vdash L_{Seq_1}(k,n)\text{[6]}
$$

$$
L_{Seq_1}(k,n+1), L_{Seq_2}(k,n+1), L_{Next}(n+1,x) \vdash L_{Seq_2}(k,n).
$$

Note that these invariants not only have a non-inductive argument of the ω sort, the inductive parameter differs from the parameter used by loop (1), that is k rather than n. These properties are easily handled by Definition 7. Furthermore, while we provided a construction starting from the schematic NNF

[6] See the NNF construction of Definition 15 for the meaning of the labels.

Φ, a similar refutation can be constructed starting from Ψ albeit with much more tedious contructions for building up the instantiations, given that the sequences are not built into the schematic NNF definitions. Note that the schematic NNF Φ is dependent on the assumptions

$$L_{Top}(n{+}1,x){\vdash}L_{Seq_1}(n{+}1,n{+}1) \qquad L_{Top}(n{+}1,x){\vdash}L_{Seq_2}(n{+}1,n{+}1).$$

which is nothing more than an abbreviation of particular instances of the clauses

$$L_{Top}(n{+}1,x){\vdash}(n{+}1){=}f(\mathbf{S}(x))\ ,\ f(x){<}(n{+}1) \qquad L_{Top}(n{+}1,x){\vdash}(n{+}1){=}f(x)\ ,\ f(x){<}(n{+}1)$$

from the schematic NNF Ψ. Thus, these assumptions are trivial and show up as analytic cuts in the GAPT implementation. Such analytic cuts are precisely what the structural clause form allows one to use. Furthermore, these assumptions allow us to reorganize the instantiations in such a way as to allow the construction of a refutation without complex computational or ordering principles. Thus, we can get around some of the shortcomings of the previous methods.

In either case a unification schema can be extracted from the refutation encoding the term instantiations of the sequences $Seq_1(m,m)$, $Seq_2(m,m)$ for all $0 \leq m \leq n$. A full formalization of this refutation as well as the proof schema can be found within the examples directory of the GAPT system.

Using GAPT it is possible to extract a Herbrand sequent for a parameter n. The following is the Herbrand sequent for case $n = 1$.

$$\bigwedge_{i=\mathbf{0}}^{\mathbf{S}(\mathbf{S}(\mathbf{0}))} \bigvee_{j=0}^{1} j = f(i) \vdash \bigvee_{i=\mathbf{0}}^{\mathbf{S}(\mathbf{0})} f(i) = f(\mathbf{S}(i))$$

This can be easily generalized to the following schematic Herbrand sequent using the abbreviation

$$\mathbf{S}^n(\mathbf{0}) \equiv \overbrace{\mathbf{S}(\cdots \mathbf{S}(}^{\text{n times}}\mathbf{0})\cdots)$$

.

$$\bigwedge_{i=\mathbf{0}}^{\mathbf{S}(\mathbf{S}^n(\mathbf{0}))} \bigvee_{j=0}^{n} j = f(i) \vdash \bigvee_{i=\mathbf{0}}^{\mathbf{S}^n(\mathbf{0})} f(i) = f(\mathbf{S}(i)).$$

Note that this schematic Herbrand sequent is provable modulo the theory outlined in Definition 10, most notably Equation (3). Notice that the antecedent of the schematic Herbrand sequent has nested iterations. This is precisely the reason why previously developed methods fail to analyze this formal proof. Both the method of [15] and [19] rely on the clause set having a refutation with a simple looping structure and thus a Herbrand sequent with a simple term structure, which is impossible for this example. As we can work with whole formulas and insert propositional cuts the new formalism is flexible enough to handle a complex refutation as the one discussed here.

5 Conclusion

We introduced a schematic resolution calculus based on schematic clause forms, which serves as an improvement of the calculus introduced in [15] and provides an interactive alternative to [19]. We intend to use the developed calculus for two major applications: first of all, it can be used to define a so-called inessential cut-normal form as in [19]. Having obtained a normal form, proof analysis (extraction of schematic Herbrand sequents and expansion trees) comes within reach. Moreover, by using globalization variables for unification we can simplify proof analysis with the method CERES considerably. Indeed, instead of constructing a proof projection for every instance of the characteristic formula (as described in [19]), we can construct one proof projection and instantiate the whole proof. Therefore, we can use a method based on the extraction of expansion trees from proof projections and hence omitting proof normalization (i.e. the construction of an inessential cut normal form), similar to the method defined in [18].

Furthermore, we believe that, independent of its application to proof analysis, the schematic resolution calculus defined in this paper provides a simple and powerful tool for interactive proof construction in inductive reasoning.

References

1. Vincent Aravantinos, Ricardo Caferra, and Nicolas Peltier. Decidability and undecidability results for propositional schemata. *Journal of Artificial Intelligence Research*, 40(1):599–656, January 2011.
2. Vincent Aravantinos, Mnacho Echenim, and Nicolas Peltier. A resolution calculus for first-order schemata. *Fundamenta Informaticae*, 2013.
3. Matthias Baaz, Uwe Egly, and Alexander Leitsch. Normal form transformations. In *Handbook of Automated Reasoning (in 2 volumes)*, pages 273–333. 2001.
4. Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter, and Hendrik Spohr. Ceres: An analysis of Fürstenberg's proof of the infinity of primes. *Theoretical Computer Science*, 403(2-3):160–175, August 2008.
5. Matthias Baaz and Alexander Leitsch. Cut-elimination and redundancy-elimination by resolution. *Journal of Symbolic Computation*, 29:149–176, 2000.
6. George Boolos. Don't eliminate cut. *Journal of Philosophical Logic*, 13(4):373–378, 1984.
7. James Brotherston. Cyclic proofs for first-order logic with inductive definitions. In *Tableaux'05*, volume 3702 of *Lecture Notes in Comp. Sci.*, pages 78–92. 2005.
8. James Brotherston and Alex Simpson. Sequent calculi for induction and infinite descent. *Journal of Logic and Computation*, 21(6):1177–1216, 2010.
9. David Cerna. A tableaux-based decision procedure for multi-parameter propositional schemata. In *Intelligent Computer Mathematics*, volume 8543 of *Lecture Notes in Comp. Sci.*, pages 61–75. Springer, 2014.
10. David M. Cerna. *Advances in schematic cut elimination.* PhD thesis, Technical University of Vienna, 2015. http://media.obvsg.at/p-AC12246421-2001.
11. David M. Cerna and Alexander Leitsch. 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*, pages 241–256, 2016.

12. David M. Cerna and Michael Lettmann. Integrating a global induction mechanism into a sequent calculus. In *TABLEAUX'17*, Lecture Notes in Comp. Sci., Sept. 2017.
13. David M. Cerna and Michael Lettmann. Towards a clausal analysis of proof schemata. In *SYNASC'17*, IEEE Xplorer. IEEE, Sept. 2017.
14. Cvetan Dunchev. *Automation of cut-elimination in proof schemata*. PhD thesis, Technical University of Vienna, 2012.
15. Cvetan Dunchev, Alexander Leitsch, Mikheil Rukhaia, and Daniel Weller. Cut-elimination and proof schemata. In *Logic, Language, and Computation*, pages 117–136, 2013.
16. Gabriel Ebner, Stefan Hetzl, Giselle Reis, Martin Riener, Simon Wolfsteiner, and Sebastian Zivota. *System Description: GAPT 2.0*, pages 293–301. Springer International Publishing, Cham, 2016.
17. Stefan Hetzl, Alexander Leitsch, Daniel Weller, and Bruno Woltzenlogel Paleo. Herbrand sequent extraction. In *Intelligent Computer Mathematics*, pages 462–477. Springer, 2008.
18. Alexander Leitsch and Anela Lolic. Extraction of expansion trees. *The Journal of Automated Reasoning*, 2018. DOI: 10.1007/s10817-018-9453-9.
19. Alexander Leitsch, Nicolas Peltier, and Daniel Weller. CERES for first-order schemata. *J. Log. Comput.*, 27(7):1897–1954, 2017.
20. Raymond Mcdowell and Dale Miller. Cut-elimination for a logic with definitions and induction. *Theoretical Computer Science*, 232, 1997.
21. Dale A. Miller. A compact representation of proofs. *Studia Logica*, 46(4):347–370, 1987.
22. V.P. Orevkov. Proof schemata in Hilbert-type axiomatic theories. *Journal of Soviet Mathematics*, 55(2):1610–1620, 1991.
23. Gaisi Takeuti. *Proof Theory*, volume 81 of *Studies in logic and the foundations of mathematics*. American Elsevier Pub., 1975.
24. Jannik Vierling. Cyclic superposition and induction. Master's thesis, Technical University of Vienna, 2018.