A Resolution Calculus for Recursive Clause Sets

David M. Cerna, Alexander Leitsch, and Anela Lolic

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. We provide an example based on an important theoretical case and a procedure for constructing an inessential cut normal form schema.

Keywords: Proof Schemata, Resolution calculus, Inductive proofs, Expansion trees, Recursion

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 quite close to the type of proof schemata we will discuss in this paper. 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. Moreover, Herbrand’s theorem can be extended to an expressive fragment of proof schemata, that is those formalizing k-induction [15, 18]. Unfortunately, the construction of recursively defined 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. We provide an example based on an important theoretical case and a procedure for constructing an inessential cut normal form schema.

Keywords: Proof Schemata, Resolution calculus, Inductive proofs, Expansion trees, Recursion

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 quite close to the type of proof schemata we will discuss in this paper. 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. Moreover, Herbrand’s theorem can be extended to an expressive fragment of proof schemata, that is those formalizing k-induction [15, 18]. Unfortunately, the construction of recursively defined

*This research is supported by FWF project P28789-N32
†This work is partially supported by ...
refutations is a high hurdle to overcome when applying the method. In previous work [18] a superposition calculus for certain types of formulas was used for the construction of refutation schemata, but it only works for a weak fragment of arithmetic\(^1\) and is hard to use interactively.

Historically, it was the schematic construction of object level concepts that formed the basis for proof schemata, as described in this paper. These foundational ideas evolved from work 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. \(\lor\)-iteration and \(\land\)-iteration. Their investigations led to a tableau-based decision procedure for the satisfiability of a monadic\(^2\) 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 [18].

Nonetheless, these results inspired investigations into the use of schemata as an alternative formalization of induction for proof analysis and proof transformation. However, this is not the first alternative formalization of induction with respect to Peano arithmetic [23]. On the other hand, all existing examples [7, 8, 20], to the best of our knowledge, lack a proof normal form or subformula-like property\(^3\). By the term “subformula-like”, we mean that the proof is not necessarily fully analytic\(^4\), but may contain quantifier-free cuts (note that for proofs of this form Herbrand sequents can be constructed). 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 [16, 23] and expansion trees [21], cannot directly 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 [18] finite representations of sequences of Herbrand sequents are constructed, so-called Herbrand systems. Of course, such objects do not describe finite set 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, 18] due to the restriction placed on proof

---

\(^1\) The system is \(\Sigma_1\) induction complete while important formulations of the infinitary pigeonhole principle [10] are \(I_2\). See the soon to be completed Master Thesis of Jannik Vierling “Cyclic Superposition and Induction”.

\(^2\) In this fragment the use of schematic constructors is restricted to one free parameter per formula.

\(^3\) subformula property means that every formula occurring in the proof is a subformula of a formula occurring in the end sequent.

\(^4\) a proof is analytic when it fulfills the subformula property.
schema construction, see [10]. Lifting these restrictions is non-trivial. Our goal is to design a calculus which easily allows one to relax the restrictions.

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 pigeon hole principle [11]. An improved version of this cut-elimination method has been introduced in [18], 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.

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 a second parameter. Moreover, the method improves the approach in [18] 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, 18]. We provide schematic inessential cut normal forms like in [18] and give a sketch of a method extracting schematic expansion trees by an approach similar to [17, 19]. We think 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 being that 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, 18].

We work in a two sorted logic were the $\iota$ sort is the first-order term sort and $\omega$ is the numeric sort, i.e. terms constructed from the alphabet $\{0, s(\cdot)\}$. We are limited to predicate symbols with at most a single numeric index, a so-called parameter. Both sorts have an associated countable set of variables, $V_\iota$ and $V_\omega$. We refer to special constants of the $\omega$ sort as parameters.

We extend the alphabet of the $\iota$ sort by defined function symbols, i.e. primitive recursively defined functions. Analogously, we allow defined predicate symbols to build formula schemata. Defined symbols will be denoted by $\hat{}$, i.e. $\hat{P}$. We assume a set of convergent rewrite rules $\mathcal{E}$ (equational theory) for defined symbols. The underlying equations of $\mathcal{E}$ are of the form $\hat{f}(\bar{t}) = E$, where $\bar{t}$ contains no defined symbols, and either $\hat{f}$ is a defined function symbol and $E$ is a term or $\hat{f}$ is a defined predicate symbol and $E$ is a formula schema.

---

See Orevkov’s proof [22] or Boolos’ proof [6].
Definition 1 (LKE). Let $\mathcal{E}$ be an equational theory. LKE is an extension of LK by the $\mathcal{E}$ inference rule $\frac{S(t)}{S(t')}$. Let us define a proof schema $\mathcal{C}$ containing components and the end sequent of $\mathcal{C}$ for $\mathcal{E} \models t = t'$.

A schematic sequent is a pair of multisets of formula schemata $\Delta$, $\Pi$ denoted by $\Delta \vdash \Pi$. We will denote multisets of formula schemata by upper-case Greek letters. Let $S(\pi)$ be a sequent and $\pi$ a vector of free variables, then $S(\pi)$ denotes $S(\pi)$ where $\pi$ is a vector of terms of appropriate type. We refer to $\varphi(\pi(n))$, where $S(\pi)$ is a schematic sequent and $\varphi$ a proof symbol as a proof link. The sequent calculus LKS consists of the rules of LKE extended by proof links. Proof links act as place holders for proofs and during evaluation they are replaced by the proof corresponding to the proof symbol of the link (see [18] for details concerning evaluation).

Definition 2 (Proof schema component). Let $\psi$ be a proof symbol, $n$ a parameter, $\pi$ an LKS-proof without links and not containing $n$ and $\nu$ an LKS-proof containing links and containing $n$. Then $\mathcal{C} = (\psi, \pi, \nu)$ is a proof schema component. The end sequents of $\pi$ and $\nu$ are $S\{n \leftarrow 0\}$ and $S\{n \leftarrow s(n)\}$, respectively. The end sequent of $\mathcal{C}$ is $S$ and $\text{sym}(\mathcal{C}) = \psi$.

Definition 3. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be distinct proof schema components with end sequents $S_1$ and $S_2$, respectively. We say $\mathcal{C}_1 >^* \mathcal{C}_2$ if $\mathcal{C}_2$ does not link to $\mathcal{C}_1$ and all links to $\mathcal{C}_1$ or $\mathcal{C}_2$ are of the form: $\text{sym}(\mathcal{C}_1), \hat{l}$ or $\text{sym}(\mathcal{C}_2), \hat{i}$ for $k \in [1, 2]$ s.t. $\mathcal{V}(\hat{l}) \subseteq \{n\}$ and $\hat{i}$ is a sequence of terms of the $i$ sort. Let $\Psi$ be a set of proof schema components containing $\mathcal{C}_1$ and $\mathcal{C}_2$. We say $\mathcal{C}_1 > \mathcal{C}_2$ if $\mathcal{C}_1 >^* \mathcal{C}_2$ and $\mathcal{C}_1 >^* \mathcal{D}$ holds for all proof schema components $\mathcal{D}$ in $\Psi$ with $\mathcal{C}_2 >^* \mathcal{D}$.

Definition 4 (Proof schema [15]). Let $\mathcal{C}_1, \ldots, \mathcal{C}_m$ be distinct proof schema components and the end sequent of $\mathcal{C}_1$ be $S$. $\Psi = \langle \mathcal{C}_1, \ldots, \mathcal{C}_m \rangle$, s.t. $\mathcal{C}_1 > \ldots > \mathcal{C}_m$, is a proof schema with end sequent $S$.

Example 1 (Proof schema). Let us define a proof schema $\langle (\phi, \pi, \nu) \rangle$ with end sequent $P(0), F(n) \vdash P(n + 1)$ using the definitions
\[
\mathcal{E} = \left\{ \begin{array}{l}
F(0) = P(0) \rightarrow P(1) \\
F(n + 1) = P(n + 1) \rightarrow P(n + 2) \wedge F(n) 
\end{array} \right\}.
\]

$\pi$ and $\nu$ are as follows:

\[
\frac{P(0) \vdash P(0)}{P(0), P(0) \vdash P(1)} \ldots \frac{P(0), P(0) \vdash P(1)}{P(0), P(0) \vdash P(n + 1)} \ldots \frac{P(0), P(0) \vdash P(n + 1)}{P(0), P(n + 1) \vdash P(n + 2)}
\]

where $\nu_1$ and $\nu_2$ are

\[
\frac{P(0), P(n) \vdash P(n + 1)}{P(0), P(n) \vdash P(n + 1)}
\]

Now that we have defined proof schema we can move onto the concepts important for this work, the characteristic formula schema and the proof projection.
The concepts are defined in great detail in [18], we only provide a succinct introduction to facilitate understanding of our work. The main issue separating the schematic version of the concepts from the first-order version is the cut-status of the formula occurrences in a given proof (i.e. whether a given formula occurrence is a cut-ancestor, or not). Not only can a cut-ancestor occur in the currently evaluated proof, but cut-ancestor status can be passed down from another proof through a link. Therefore we need some machinery to track the cut-status of formula occurrences through links. Thus, the schematic characteristic formula schema, as well as the proof projection, require cut configuration information to be passed through the links.

**Definition 5 (schematic characteristic formula [18]).** Let $\pi$ be an LKS-proof $\Xi = \Omega \cup C$ and $\rho$ an inference in $\pi$. We define the characteristic formula at $\Theta_\rho(\pi, \Omega)$ inductively:
- If $\rho$ is an axiom $S$ then $\Theta_\rho(\pi, \Omega) = F$, where $F$ is the formula from $S \cap \Xi$.
- If $\rho$ is a proof link of the form $(\psi, t_1, \cdots, t_n)$ with conclusion $S$ with $\Omega'$ as a set of formula occurrences from $\Xi \cap S$ and $\Theta_\rho(\pi, \Omega) = \Theta_\psi(\pi, \Omega')(t_1, \cdots, t_n)$.
- If $\rho$ is a unary rule with immediate predecessor $\rho_1$ of, then $\Theta_\rho(\pi, \Omega) = \Theta_{\rho_1}(\pi, \Omega)$.
- If $\rho$ is a binary rule with immediate predecessors $\rho_1, \rho_2$ we distinguish two cases.
  - If the auxiliary formulas of $\rho$ are $\Xi$-ancestors, then $\Theta_\rho(\pi, \Omega) = \Theta_{\rho_1}(\pi, \Omega) \lor \Theta_{\rho_2}(\pi, \Omega)$.
  - Otherwise $\rho$ are $\Xi$-ancestors, then $\Theta_\rho(\pi, \Omega) = \Theta_{\rho_1}(\pi, \Omega) \lor \Theta_{\rho_2}(\pi, \Omega)$.

Finally, define $\Theta(\pi, \Omega) = \Theta_{\rho_0}(\pi, \Omega)$, where $\rho_0$ is the last inference of $\pi$, and $\Theta(\pi) = \Theta(\pi, \emptyset)$. We lift this definition to $\psi$: let $n$ be a variable of type $\omega$ not occurring in $\psi$, then define $\Theta(\psi) = \Theta_{\psi, \Omega}(n)$.

**Definition 6 (proof projection [18]).** Let $\pi$ be an LKS-proof of $\Gamma \vdash \Delta, \Omega$ a configuration for $\pi$ and $F = \Theta(\pi, \Omega)$. We define an LKS-proof $\pi^*(\Omega)$ of $\Gamma' \vdash \Delta', F$, where $\Gamma', \Delta'$ denote the subsequences of the formulas of $\Gamma, \Delta$ respectively that do not occur in $\Omega$, inductively, according to the inference $\rho$ at root level in $\pi$. First of all, we denote with $\Gamma'', \Delta''$ the subsequences of the formulas of $\Gamma, \Delta$ respectively that occur in $\Omega$. We assume that all proof symbols $\psi$ are mapped to pairwise distinct proof symbols $\psi'$, ordered as the symbols $\psi$.
- If $\rho$ is an axiom then so is the sequent $\Gamma', \Gamma'' \vdash \Delta', \Delta''$ and by definition $F = \Theta(\pi, \Omega) = \bigvee \neg \Gamma'' \lor \Delta''$ (note that $\Gamma', \Delta = A \vdash A$ and that one of $\Gamma', \Gamma''(\Delta', \Delta'')$ is empty). The proof $\pi^*(\Omega)$ is constructed as follows:
  \[
  \frac{\Gamma \vdash \Delta}{\Gamma' \vdash \Delta', \neg \Gamma'', \Delta''} \quad \because : r
  \frac{\Gamma' \vdash \Delta', F}{\Gamma'' \vdash \Delta'} \quad \lor : r
  \]
- If $\rho$ is a proof link of the form $(\psi t_1, \cdots, t_n)$ then by definition $F$ is of the form $\Theta_\psi(\Omega')(t_1, \cdots, t_n)$, for some configuration $\Omega'$. Due to the rules defining $\Theta_\psi(\Omega')(t_1, \cdots, t_n)$.\footnote{Extract the common part of both sequents.}
\[ \Theta_{\psi, \varphi}, \Theta_{\psi, \varphi}(t_1, \ldots, t_\alpha) \{n \leftarrow 0\} \downarrow \text{ and } \Theta_{\psi, \varphi}(t_1, \ldots, t_\alpha) \{n \leftarrow n + 1\} \downarrow \text{ are of the form } \Theta(\pi, \Omega') \downarrow \text{ and } \Theta(\varphi, \Omega') \downarrow \text{ respectively, where } (\psi, \pi, \varphi) \text{ is a proof schema component and } \varphi \text{ is the substitution } [x_1 \leftarrow t_1, \ldots, x_\alpha \leftarrow t_\alpha]. \] This implies that \( \Theta_{\psi, \varphi}(t_1, \ldots, t_\alpha) \{n \leftarrow 0\} \downarrow = \Theta(\pi, \Omega') \downarrow \text{ and } \Theta_{\psi, \varphi}(t_1, \ldots, t_\alpha) \{n \leftarrow n + 1\} \downarrow = \Theta(\varphi, \Omega') \downarrow. \] Then the formula \( F \) is denoted by \( \hat{f} \) associated with two rules of the form \( \hat{f}(0) \rightarrow \Theta(\pi, \Omega') \) and \( \hat{f}(k + 1) \rightarrow \Theta(\varphi, \Omega') \). We let \( \pi^*(\Omega) = \hat{\psi}'(t_1) \), where the rules defining \( \hat{\psi}' \) are \( \hat{\psi}'(0) \rightarrow \pi^*(\Omega) \) and \( \hat{\psi}'(k + 1) \rightarrow \pi^*(\Omega') \).

- Assume that \( \rho \) is a unary rule. Let \( \pi_1 \) be the immediate predecessor of \( \pi \) and let \( \Gamma_1 \vdash \Delta_1 \) be the end-sequent of \( \pi_1 \). By definition, \( F = \Theta(\pi, \Omega) = \Theta(\pi_1, \Omega) \). If the inference \( \rho \) occurs on a formula in \( \Omega \), we have \( \Gamma_1'' = \Gamma'' \) and \( \Delta_1'' = \Delta'' \), hence \( \pi^*(\Omega) = \pi_1^*(\Omega) \). Otherwise we can apply the inference \( \rho \) on \( \Gamma_1'' \vdash \Delta_1'', \) \( F \) yielding \( \Gamma'' \vdash \Delta'', \) \( F \).

- Assume that \( \rho \) is a binary rule. Let \( \pi_1, \pi_2 \) be the immediate predecessors of \( \pi \) and let \( \Gamma_i \vdash \Delta_i \) be the end-sequent of \( \pi_i \) \( (i = 1, 2) \). Let \( F_i = \Theta(\pi_i, \Omega) \) and let \( \Gamma_i', \Delta_i' \) be the subsequences of \( \Gamma_i, \Delta_i \) that do not occur in \( \Omega \). We distinguish two cases.

  - If the auxiliary formulas of \( \rho \) are \( \Omega \)-ancestors, then \( F = \Theta(\pi, \Omega) = \Theta(\pi_1, \Omega) \land \Theta(\pi_2, \Omega) = F_1 \land F_2 \). Furthermore, \( \Gamma' \) and \( \Delta' \) are the concatenations of \( \Gamma_1', \Gamma_2' \) and \( \Delta_1', \Delta_2' \). Thus \( \Gamma_i'' \vdash \Delta_i', \) \( F \) can be derived from \( \Gamma_i'' \vdash \Delta_i, \) \( F_1 \) and \( \Gamma_i'' \vdash \Delta_i, \) \( F_2 \) by the rule \( \land : r \).

  - Otherwise \( \rho \) are \( \Omega \)-ancestors and \( F = \Theta(\pi, \Omega) = \Theta(\pi_1, \Omega) \lor \Theta(\pi_2, \Omega) = F_1 \lor F_2 \). We can apply the rule \( \lor : R \) yields \( \Gamma'' \vdash \Delta', \) \( F_1 \lor F_2 \), i.e., \( \Gamma'' \vdash \Delta', \) \( F \).

The construction terminates since there are finitely many proof links. It is easy to check that all the inferences are valid in \text{LKS} \) and that the end-sequent of \( \pi^*(\Omega) \) is indeed \( \Gamma'' \vdash \Delta', \) \( F \). A proof projection schema \( \Phi_p \) of a proof schema \( \Phi \) is a proof schema where every \text{LKS}-derivation in \( \Phi \) is replaced with its proof projection and the end-sequent of \( \Phi_p \) is the end-sequent of \( \Phi \) composed with \( \vdash F \) where \( F \) is the characteristic formula schema of \( \Phi \).

In the next section we provide example characteristic formula Extracted from the proof schema introduced in [18].

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 [18] for schematic cut-elimination. Our approach here is based on the method of
Definition 7 (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 labeled 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 $\Phi(F, L_F) = \{L_F(\overline{y}), A \vdash \}$. 

- $F = F_1 \land F_2$. 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 $\{L_F(\overline{y}) \vdash L_{F_1}(\overline{y}_1), L_{F_2}(\overline{y}_2)\} \cup \Phi(F_1, L_{F_1}) \cup \Phi(F_2, L_{F_2})$.

- $F = F_1 \lor F_2$. 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 8 (definitional form). Let $\Phi(F, L_F)$ be the definitional clause set of $F$. Then the set of clauses $\text{CL}(F, L_F) : \{\vdash L_F(\overline{y})\} \cup \Phi(F, L_F)$ is called the definitional form of $F$.

Proposition 1. Let $F$ be a formula in NNF. Then

1. $F$ and $\text{CL}(F, L_F)$ are sat-equivalent.

2. Let $c$ be a clauses of $\text{CL}(F, L_F)$ and $LR(c) = \Delta \vdash \Pi$ be $c$ where each label is replaced by the formula it represents. Then $F, \Delta \vdash \Pi$ is provable in LK.

Proof. simple. An LK proof $\varphi(F)$ of $\text{CL}(F) \vdash F$ is computable from $\text{CL}(F, L_F)$ in polynomial time.

Definition 9 (schematic clause form). Let $\Psi$ be a schematic NNF definition with defined symbols $\hat{F}_1, \ldots, \hat{F}_k$ such that $\hat{F}_1 > \hat{F}_2 > \cdots > \hat{F}_k$ s.t. 

$\hat{F}_i(\overline{\gamma}_i, 0) = G_i(\overline{\gamma}_i)$, 

$\hat{F}_i(\overline{\gamma}_i, n + 1) = H_i(\overline{\gamma}_i, \hat{F}_i(\overline{\gamma}_i, n))$,

where the $G_i(\overline{\gamma}_i)$ and $H_i(\overline{\gamma}_i, \hat{\top})$ are formulas in NNF not containing $\hat{F}_j$ for $j < i$.

Then the structural form $\text{CL}(\Psi)$ of $\Psi$ is defined as $\text{CL}(\Psi, 0), \text{CL}(\Psi, n + 1))$ for

$\text{CL}(\Psi, 0) = \{L_F(\overline{\gamma}, 0)\} \cup \Phi(G_1(\overline{\gamma}_1), L_{F_1}) \cup \cdots \cup \Phi(G_k(\overline{\gamma}_k), L_{F_k})$,

$\text{CL}(\Psi, n + 1) = \{L_F(\overline{\gamma}, n + 1)\} \cup \Phi(H_1(\overline{\gamma}_1, L_{F_1}(\overline{\gamma}_1, n)), L_{F_1}) \cup \cdots \cup \Phi(H_k(\overline{\gamma}_k, L_{F_k}(\overline{\gamma}_k, n)), L_{F_k})$. 
Note that, for any schematic NNF-definition $\Psi$, $\text{CL}(\Psi) : (\text{CL}(\Psi, 0), \text{CL}(\Psi, n + 1))$ is a pair of finite set of clauses.

Below we are defining a resolution calculus for schematic clause sets.

**Definition 10 (schematic clause).** Let $C : P_1, \ldots, P_n \vdash Q_1, \ldots, Q_\beta$ be a sequent where the $P_i$ and $Q_j$ are schematic atoms and there is only one parameter occurring in $C$. Then $C$ is called a schematic clause.

**Remark 1.** Schematic clauses are just atomic sequents consisting of schematic atoms where schematically defined terms may occur. Note that this formalism is much simpler than that in [15] where schematic clauses required additional recursive definitions on the clause level.

**Definition 11 (resolution deduction).** Let $C$ be a set of schematic clauses over a parameter $n$. Then, for every clause $C \subset C'$ (where $C'$ is a variant of $C$) is a resolution deduction of $C$ from $C$. Assume that $\delta_1$ is a resolution deduction of $C_1 : \Gamma \vdash \Delta, A_1, \ldots, A_\alpha$ from $C$ and $\delta_2$ is a resolution deduction of $C_2 : B_1, \ldots, B_\beta \Pi \vdash \Lambda$ from $C$ such that $\delta_1$ and $\delta_2$ are variable disjoint (but both may contain the parameter $n$). Assume further that $\Theta$ is a unifier of $\{A_1, \ldots, A_\alpha, B_1, \ldots, B_\beta\}$. Then

$$
\frac{(\delta_1) \quad (\delta_2)}{C : \Gamma \Theta, \Pi \Theta \vdash \Delta \Theta, \Lambda \Theta} \Theta
$$

is a resolution deduction of $C$ from $C$.

**Remark 2.** The concept of resolution deduction almost coincides with the usual one. The only exception is the unification principle which is defined over schematic terms and is undecidable (see [18]). Hence these definitions strongly differ from this in [15] where resolution terms were used.

For schematic resolution deductions we need inductive definitions. Note that for schematic clause sets $C$ 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 $\Delta^*$ and define a partial order $\triangleleft$ on them.

**Definition 12 (schematic resolution deduction).** Let $\Delta$ be a finite subset of $\Delta^*$ such that there exists a $\delta_0 \in \Delta$ with $\delta_0 > \delta'$ for all $\delta' \in \Delta$ such that $\delta' \neq \delta_0$. A finite set of tuples $R : \{(\delta, \rho(\delta, n, 0), \rho(\delta, n, m + 1)) \mid \delta \in \Delta\}$ is called a schematic resolution deduction from a finite set of schematic clauses $C$ over the parameter $n$ if the following conditions hold for every $\delta \in \Delta$:

There exists a (possibly empty) finite set of clauses $C(\delta)$ and a clause $D(\delta)$ such that

1. $\rho(\delta, n, 0)$ is a resolution deduction of $D(\delta) \{m \leftarrow 0\}$ from $C \cup C(\delta)$,
(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 \( \{m \leftarrow n + \alpha\} \) for some numeral \( \alpha \)) of a clause \( D(\delta') \) for a \( \delta' \in \Delta \) with \( \delta > \delta' \) such that the conditions (1) and (2) hold for \( \delta' \).

If \( D(h_0) \) is the empty clause \( \vdash \) we call \( R \) a resolution refutation schema of \( \mathcal{C} \).

Note that the concept is well defined as there are minimal elements in \( \Delta \) 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 \( \beta, \gamma \) 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 \( D(\delta')[m \leftarrow n + \alpha] \) by the deductions \( \rho(\delta', n, n + \alpha) \).

**Definition 13.** Let \( \Psi \) be a schematic NNF-definition and \( \text{CL}(\Psi, 0) \), \( \text{CL}(\Psi, n + 1) \) be the corresponding set of clauses. Then the pair \( (R_b, R_s) \), where \( R_b \) is a resolution refutation schema of \( \text{CL}(\Psi, 0) \) and \( R_s \) is a resolution refutation schema of \( \text{CL}(\Psi, n + 1) \), is called a refutation schema for \( \Psi \).

**Example 2.** Consider the schematic formula \( P(a) \land (\neg P(x) \lor P(f(x))) \land \neg P(f^n(a)) \), a characteristic formula (in fact a simplified version under tautology elimination and subsumption) resulting from a schematic cut-elimination problem (see [18], Example 3.6). The corresponding clause set is \( \mathcal{C}_n = \{P(a), \neg P(x) \lor P(f(x)), \neg P(f^n(a))\} \) with the following schematic NNF formalization \( \Psi = \)

\[
\begin{align*}
\hat{F}(0, x) & = P(a) \\
\hat{F}(n + 1, x) & = P(a) \land (\neg P(x) \lor P(f(x))) \\
\hat{G}(n) & = \neg P(f(n, a)) \\
\hat{H}(0, x) & = \hat{F}(0, x) \land \hat{G}(0) \\
\hat{H}(n + 1, x) & = \hat{F}(n + 1, x) \land \hat{G}(n + 1) \\
R(\hat{f}) & = \begin{cases} \hat{f}(0, x) \rightarrow x \\ \hat{f}(n + 1, x) \rightarrow f(\hat{f}(n, x)) \end{cases}
\end{align*}
\]

with \( \hat{H} > \hat{F}, \hat{H} > \hat{G} \)

\[
\text{CL}(\Psi, 0) = \{\vdash L_H(0, x)\} \cup \{L_H(0, x) \vdash L_F(0, x) , L_H(0, x) \vdash L_G(0), L_F(0, x) \vdash P(a), L_G(0), P(\hat{f}(0, a)) \vdash \}
\]

\[
\text{CL}(\Psi, n + 1) = \{\vdash L_H(n + 1, x)\} \cup \{L_H(n + 1, x) \vdash L_F(n + 1, x) , \}
\]

\[
L_H(n + 1, x) \vdash L_G(n + 1) , L_G(n + 1), P(f(n + 1, a)) \vdash , L_F(n + 1, x) \vdash P(a), L_F(n + 1, x) , P(x) \vdash P(f(x)) \} \]
We define a refutation schema \((\mathcal{R}_b, \mathcal{R}_a)\) of \(\Psi\).
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 \(\phi_1\) be
\[
\frac{\vdash L_H(0, x) \quad \vdash L_H(0, y) \quad \vdash L_G(0)}{P(\tilde{f}(0, a)) \vdash} y \leftarrow x
\]
and \(\phi_2\) be
\[
\frac{\vdash L_H(0, u) \quad \vdash L_H(0, v) \quad \vdash L_F(0, v) \quad \vdash L_F(0, u) \quad \vdash P(a)}{P(\tilde{f}(0, a)) \vdash} v \leftarrow u
\]
Then \(\rho(\delta^b, n, 0)\) is defined as
\[
\frac{\phi_1 \quad \phi_2}{\vdash} \varnothing
\]
Clearly \(\rho(\delta^b, n, m)\) is an LK-refutation of \(\text{CL}(\Psi, 0)\).

Now we construct a schematic resolution derivation \(\mathcal{R}_1 = \{\delta, \rho(\delta, n, 0), \rho(\delta, n, m + 1)\}\) where \(D(\delta) = P(\tilde{f}(m, a))\).

\[
\rho(\delta, n, 0) = \quad \frac{\vdash L_H(n + 1, y) \quad \vdash L_H(n + 1, x) \quad \vdash L_F(n + 1, x)}{P(\tilde{f}(n, a)) \vdash} y \leftarrow x \quad L_F(n + 1, x) \vdash P(a) \quad z \leftarrow z
\]

Note that \(D(\delta)[m \leftarrow 0] = P(a)\).

\[
\rho(\delta, n, m + 1) = \quad \frac{\vdash L_H(n + 1, y) \quad \vdash L_H(n + 1, x) \quad \vdash L_F(n + 1, x) \quad \vdash P(\tilde{f}(m, a))}{P(\tilde{f}(n + 1, a)) \vdash} y \leftarrow x \quad L_F(n + 1, x) \vdash P(a) \quad z \leftarrow z
\]

for \(C(Z, n, m) = L_F(n + 1, z), P(z) \vdash P(\tilde{f}(z))\).

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

Finally we define the resolution refutation schema
\[
\mathcal{R}_s: \{\delta^s, \rho(\delta^s, n, 0), \rho(\delta^s, n, m + 1)\} \cup \mathcal{R}_1
\]
where \(\delta^s > \delta\) and \(\rho(\delta^s, n, 0) = \rho(\delta^s, n, m + 1)\) =
\[
\frac{\vdash L_H(n + 1, x) \quad \vdash L_H(n + 1, y) \quad \vdash L_G(n + 1)}{P(\tilde{f}(n + 1, a)) \vdash} y \leftarrow x \quad L_G(n + 1) \vdash P(f(n + 1, a)) \vdash \varnothing
\]
We have shown that $\mathcal{R}_\alpha$ is a refutation schema of $\text{CL}(\psi, n+1)$ and therefore $(\mathcal{R}_\theta, \mathcal{R}_\alpha)$ is a refutation schema of $\psi$.

When we apply a total unification $\theta$ to a resolution refutation $\rho$ 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. Therefore, turning a resolution refutation schema into an LK-refutation schema is vital to the computation of the schematic CERES normal form in the next section. 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 $\delta$ sets of variables, so-called globalization variables, $X^\delta_l(0)$ and $X^\delta_r(m+1)$.

We need these variables of type $\mathbb{N} \rightarrow \iota$ to avoid conflicts in variable substitutions in substituting the parameter $m$ by a numeral.

**Definition 14 (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 $\delta$ be a minimal element in $\Delta$. 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

$$\eta(\delta, n, 0) = \{x_1 \leftarrow X^\delta_l(0), \ldots, x_\alpha \leftarrow X^\delta_r(0)\}.$$ 

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

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

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

$$\mathcal{C}(\delta) = \{D(\delta_1)\{m \leftarrow n + \alpha_1\}, \ldots, D(\delta_l)\{m \leftarrow n + \alpha_l\}\}$$

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 $\rho(\delta, n, 0)\eta(\delta, n, 0)$. Then

$$\Theta(\delta, n, 0) = \{\theta(\delta, n, 0)\} \cup \Theta(\delta_1, n, n + \alpha_1)\eta(\delta, n, 0)\theta(\delta, n, 0) \cup \ldots \cup \Theta(\delta_l, n, n + \alpha_l)\eta(\delta, n, 0)\theta(\delta, n, 0).$$

Similarly we obtain

$$\Theta(\delta, n, m+1) = \Theta(\delta, n, m) \cup \{\theta(\delta, n, m+1)\} \cup \Theta(\delta_1, n, n + \alpha_1)\eta(\delta, n, m+1)\theta(\delta, n, m+1) \cup \ldots \cup \Theta(\delta_l, n, n + \alpha_l)\eta(\delta, n, m+1)\theta(\delta, n, m+1).$$
where $\theta(\delta, n, m + 1)$ is a total unifier of $\rho(\delta, n, m + 1)\eta(\delta, n, m + 1)$.

Remark 3. Note that 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 $(R_b, R_s)$ be the refutation schema defined in Example 2. The variables in $\rho(\delta^b, n, 0)$ are $\{y \leftarrow x, v \leftarrow u, w \leftarrow u\}$. Therefore

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

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

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

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

For $\rho(\delta, n, 0)$ we obtain

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

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

Finally we obtain

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

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

Finally we obtain

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

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

4 Schematic Normal Forms and Applications

The CERES method for first order logic [5] produces a so called Atomic Cut Normal Form (ACNF) by taking a resolution refutation of a proof’s characteristic clause set, grounding it, and than attaching proof projections to the leaves of the grounded refutation. In the schematic case, where the resolution refutation is recursively defined, grounding as well as attaching of the projections has not been possible without first instantiating the refutation [11, 15], and thus construction of a normal form has not been part of the schematic CERES method. Instead, the focus of previous versions of Schematic CERES has been the construction of so called Herbrand Systems a generalizations of Herbrand sequents.

In [18], the problems concerning the construction of a normal form were partially solved by introducing what has been referred to as an inessential cuts
normal form, i.e. cuts without quantifiers are allowed. The method introduced in [15] is based on the construction of an ACNF by techniques similar to the ones found in [5], though the end result is not a normal form, but rather a variation of what is now referred to as a Herbrand System as defined in [18]. While an Inessential cut normal form allows for the construction of a proof, structured as in Figure 1, a direct construction of a proof of $F_1, \ldots, F_\alpha \vdash$ was not presented in [18], rather they used the superposition prover of [2] to construct the proof. Furthermore they never used this proof, only the substitution provided by the super position prover. Unfortunately, the superposition prover of [2] is quite inexpressive and therefore is not well suited for proof transformation and analysis of significant mathematic proofs nor argumentation 1.

In this section we use the resolution calculus developed in the previous section to provide a proof of the statement $F_1, \ldots, F_\alpha \vdash$ where $F_i = F_\sigma$ s.t. $\sigma \in \Theta(\delta_0, n, n + \alpha)$ of Definition 14 for $\delta_0$, the maximum symbol used in the refutation $R$ of the definitional form $CL(F, L_F)$. Note that by Definition 8 $CL(F, L_F) = \{ \vdash L_F(\vec{y}) \} \cup \Phi(F, L_F)$, and in particular, the clause $\vdash L_F(\vec{y})$ is heavily used in the schematic resolution refutation. The example from the previous section illustrates how this clause is used to construct a refutation. What we do here is to consider the refutation $R$ of $CL(F, L_F)$ but we remove every instance of $\vdash L_F(\vec{y})$ from the refutation and replace all other instances of $L_F(\vec{y})$ by $F$, that is instances of the symbol in the antecedent of clauses. This amounts to a resolution deduction using the clauses of $\Phi(F, L_F)$. The end clause of this resolution deduction derived from $R$ is $F_1, \ldots, F_\alpha \vdash$. This deduction can be further transformed into a proof $\Phi$ of the mentioned clause by propagating the substitutions through the derivation to the leaves, and thus grounding instances of the resolution rule transforming them into propositional cuts. Note that this last step also implies replacing symbols of the form $L_H(\vec{y})$ where $H$ is a subformula of $F$ by $H$ and constructing a propositional proof of the clause which we know exists by Proposition 1. We outline this procedure in this section.

Definition 15. Let $F$ be a formula in NNF and $\Phi(F, L_F)$ its definitional clause set. For a clause $C \in \Phi(F, L_F)$ we define $LR(C)$ to be the sequent derived from $C$ by replacing each formula label $L_H$ for some $H$ a subformula of $F$, by $H$ itself.

Definition 16 (Characteristic Proof Set). Let $F$ be a formula in NNF and $\Phi(F, L_F)$ its definitional clause set. We define the characteristic proof set $PS(F)$ as the set of LK-proofs such that For every clause $C \in \Phi(F, L_F)$ there exists a unique proof $\varphi \in PS(F)$ such that the end sequent of $\varphi$ is $LR(C)$.

Note that all proofs $PS(F)$ are non-schematic and propositional.

Definition 17. Let $\mathcal{R}$ be a resolution deduction schema from $\mathcal{C}$ and $c \in \mathcal{C}$ a clause of the form $\vdash P$. We call a resolution deduction schema $Re(R, C, c)$ a reduction by $c$ if every occurrence of

\[
\frac{c \sigma - c' \sigma}{c''} \not\in
\]


Definition 19 (Unification Schema Order). Let \( R \) define a mechanism to traverse the unification schema associated with \( \text{illustrated in Figure 1} \). To complete the normal form construction we need to 

This implies that such a decoration is precisely a proof of \( \Phi \) a definitional clause set results in a schematic proof of the Herbrand conjunction. 

Note that Theorem 1 tells us that every decoration of a resolution refutation \( R \) of a definitional clause set results in a schematic proof of the Herbrand conjunction. 

This implies that such a decoration is precisely a proof of \( \Phi \) in the normal form illustrated in Figure 1. To complete the normal form construction we need to define a mechanism to traverse the unification schema associated with \( R \).

Definition 19 (Unification Schema Order). Let \( R \) be a schematic resolution refutation of the definitional form \( \text{CL}(F, L_F) \) and \( \Theta(\delta_0, n, n + \alpha) \) the unification schema associated with \( R \). We define the total linear well ordering \( \prec_u \) on \( \Theta(\delta_0, n, n + \alpha) \) (\( \delta_0 \) the maximum symbol used in the refutation) s.t. for every \( \sigma_1 \in \Theta(\delta_0, n, n + \alpha) \) there exists \( \sigma_2 \in \Theta(\delta_0, n, n + \alpha) \) such that \( \sigma_2 \prec_u \sigma_1 \) and there does not exist a \( \sigma_3 \in \Theta(\delta_0, n, n + \alpha) \) s.t. \( \sigma_2 \prec_u \sigma_3 \prec_u \sigma_1 \). We will refer to such a \( \sigma_2 \) as the \( \text{pred}(\sigma_1) \). Also, there exists \( \sigma_0, \sigma' \in \Theta(\delta_0, n, n + \alpha) \) s.t. \( \sigma_0 \) is the minimal element of the ordering and \( \sigma' \) is the maximum element.

Proposition 2. For every resolution deduction schema \( R \) and associated unification schema \( \Theta(\delta_0, n, m) \) there exists a unification schema ordering \( \prec_u \).

Definition 20 (Inessential Cut Normal Form Schema). Let \( \Phi \) be a proof schema, \( n \) its free parameter, \( \Phi_p \) its projection schema, \( F \) is the characteristic formula schema of \( \Phi \), \( R \) a decoration of the refutation of the definitional form of \( F \), \( \Theta(\delta_0, n, m) \) the associated unification schema, and \( \prec_u \) its well ordering. We define the Inessential Cut Normal Form Schema \( IC(\sigma, n, \Phi_p, R, \prec_u) \) as follows:

\[
\begin{align*}
- IC(\sigma_0, n, \Phi_p, R, \prec_u) & \Rightarrow \quad \frac{\ldots \quad R \quad \ldots \quad \Phi_p \sigma_0}{\ldots \quad F \sigma_0, \Delta' \vdash \Pi, F \sigma_0} (\text{cut + } \epsilon^*) \\
- \text{otherwise,} \quad IC(\sigma, n, \Phi_p, R, \prec_u) & \Rightarrow \quad \frac{\ldots \quad IC(\text{Pred}(\sigma), n, \Phi_p, R, \prec_u) \quad \ldots \quad \Phi_p \sigma}{\ldots \quad \Delta, F \sigma, \Delta' \vdash \Pi, F \sigma} (\text{cut + } \epsilon^*)
\end{align*}
\]

Evaluating \( IC(\sigma', t, \Phi_p, R, \prec_u) \), where \( \sigma' \) is the maximum element of \( \prec_u \) and \( t \) is a parameter free numeric term, results in a proof of the sequent \( \Delta \vdash \Pi \{ n \rightarrow t \} \).
5 Conclusion

We introduced a schematic resolution calculus which serves as an improvement of the calculus introduced in [15] and provides an interactive alternative to [18]. The calculus is based on schematic clause forms (which are always finite sets of clauses) and forms the basis for the development of a schematic inessential cut normal form. Note that a schematic normal form has not been introduced so far and thus, our calculus provides a further improvement of the previously defined methods. Indeed, having defined a schematic normal form, the extraction of schematic expansion proofs comes within reach. Expansion proofs, generalizing Herbrand’s theorem to the non prenex case, provide a compact representation of proofs focusing on the instantiations of quantified variables [21]. Hence, in proof analysis expansion proofs play a vital role. We plan to investigate the extraction of schematic expansion proofs from schematic normal forms in future work, constituting a substantial progress in the analysis of schematic proofs.

References


