Cut-Elimination in Schematic Proofs and Herbrand Sequents

David Cerna
Institute of computer Languages
Vienna University of Technology
cernadavid1@logic.at

Alexander Leitsch
Institute of computer Languages
Vienna University of Technology
leitsch@logic.at

In a recent paper [8], a procedure was developed extending the first-order CERES method [4] so that it can handle cut-elimination in a schematic first-order calculus. The goal of this work was to circumvent the problems reductive cut elimination methods face when the LK calculus is extended by an induction rule. The schematic calculus can be considered a replacement for certain types of induction. In this work, we used the schematic CERES method to analyse a proof formalized in a schematic sequent calculus. The statement being proved is a simple mathematical statement about total functions with a finite range. The goal of proof analysis using the first-order CERES method [4] has been to produce an ACNF (Atomic Cut Normal Form) as the final output of cut-elimination. However, due to the complexity of the schematic method, the value and usefulness of an ACNF quickly vanishes; it is not easily parsable by humans. The Herbrand sequent corresponding to an ACNF turned out to be a valuable, compact and informative structure, which may be considered the essence of a cut-free proof in first-order logic [10]. We provide a method for extracting a schematic Herbrand sequent from the formalized proof and hint at how ,in future work we can generalize the procedure to handle a class of proofs by a suitable schematic language and calculus, and not just for a particular instance.

1 Introduction

Cut-elimination was originally introduced by G. Gentzen in [9] as a theoretical tool from which results like decidability and consistency could be proven. Cut-free proofs are computationally explicit objects from which interesting information such as Herbrand disjunctions and interpolants can be easily extracted. When viewing formal proofs as a model for mathematical proofs, cut-elimination corresponds to the removal of lemmas, which leads to interesting applications (such as one described below).

For such applications to mathematical proofs, the cut-elimination method CERES (cut-elimination by resolution) was developed [**?**]. It essentially reduces cut-elimination for a proof π to a theorem proving problem: the refutation of the *characteristic clause set* $\mathrm{CL}(\pi)$. Given a resolution refutation γ of $\mathrm{CL}(\pi)$, an essentially cut-free proof can be constructed by a proof-theoretic transformation.

It is well-known that cut-elimination in standard calculi of arithmetic, containing an induction rule, is impossible in general [15] (see also [11, 6, 14] for other approaches to inductive reasoning using induction rules). In fact, if φ is a proof of a sequent $S\colon \Gamma \to \forall x.A(x)$, where an induction rule occurs over a cut, the cut cannot be shifted over the induction rule and thus cannot be eliminated. This is not a feature of a specific cut-elimination method, but ,even in principle, inductive proofs require lemmata which cannot be eliminated. When we consider, on the other hand, the infinite sequence of proofs φ_n of $S_n\colon \Gamma \to A(n)$, every of these proofs enjoys cut-elimination. This observation motivated the investigation of a schematic CERES-method in [**?**], where schematic languages for **LK**-proofs and resolution refutations were developed. Related approaches are found in the literature on *cyclic proofs* [13, 7]. While CERES turned out to be more adequate for a description of cut-elimination than the traditional Gentzen method, the definitions of some proof objects like projections and atomic cut normal forms are highly complex and hard to construct. The mathematical content of a cut-free proof can be conveniently described by so-called

Submitted to:
CL& C 2014

Herbrand sequents, which correspond to midsequents in the prenex case [9]. Though Herbrand sequents are usually extracted from proofs after cut-elimination (or, at least, after elimination of all quantified cuts), the CERES methods makes it is possible to extract them from a resolution refutation of the characteristic clause set and the quantifier rules of the proof projections. By computing *schematic* Herbrand sequents in such a way we avoid the construction and description of complex schematic proof objects. The infinite sequence of these schematic Herbrand sequents can be considered as the result of cut-elimination on a class of inductive proofs, which cannot be obtained by Peano arithmetic with an induction rule.

In this paper we investigate a schematic version of the tape proof, representing forms of the infinite pigeon hole principle. Analogous proofs have been analyzed via functional interpretation with the aim of extracting programs [12]. The formal analysis of this proof schema by CERES requires means beyond the formalism developed in [8] (while only one free parameter is used like in [8], several bound parameters are necessary to formulate the schematic resolution refutations). For this reason we formulate most of the analysis and the construction of a schematic Herbrand sequent on the mathematical meta-level, but we indicate which language elements would be required to enable a partial automation of the proof analysis. The full development of a language extension and its implementation is left to future work. The long range aim is to develop a system for schematic cut-elimination in mathematical proofs containing induction rules, which could be used as a proof assistant.

The present work was also motivated by an application of CERES to (a formalization of) a mathematical proof: Fürstenberg's proof of the infinity of primes [1, 3]. The proof was formalized as a sequence of proofs φ_n showing that the assumption that there exist exactly n primes is contradictory. The application was performed in a semi-automated way: the characteristic clause sets $\mathrm{CL}(\varphi_n)$ were computed for some small values of n and from this, a general schema $\mathrm{CL}(\varphi_n)$ was constructed and subsequently analyzed by hand. The analysis finally showed that from Fürstenberg's proof, which makes use of topological concepts, Euclid's elementary proof could be obtained by cut-elimination.

2 The Method CERES

The purpose of this section is to quickly recapitulate fundamental aspects of the CERES method for first-order logic. Since the aim of this work is to generalize this method to the setting of first-order *schemata*, we will refer to this material later on, to emphasize the connection between the „usual" and the „schematic" CERES method. Note that the CERES method is quite different to the usual Gentzen-style way of reductive cut-elimination (for a comparison in the setting of first-order logic see [5]);

In [**?**, 5] the method CERES was defined which takes into account the global structure of a proof φ with cut; this global structure is represented as an open quantifier-free formula, the *characteristic formula* $\Theta(\varphi)$. It can be shown that $\Theta(\varphi)$ is always unsatisfiable. A resolution refutation ρ of its clause form (the *characteristic clause set*) then defines a skeleton of an ACNF[1] of the proof φ. The final step consists in inserting so-called proof projections into ρ to obtain an ACNF of φ. The steps of the method are illustrated by an example below; for the formal definitions we refer to [**?**, 5].

To simplify the understanding of the method all the premises (the auxiliary formulas of the inferences) are put in bold face, the conclusions are underlined and the ancestors of cut-formulas are marked with an asterisk in the following input proof.

[1] The abbreviation stands for Atomic Cut Normal Form. A proof formalized in the LK calculus in ACNF only has non-quantified atomic cuts. please refer to [**?**, 5] for more detail.

Now, let φ be the proof

$$\dfrac{\varphi_l \quad \varphi_r}{(\forall x)(\forall y)(P(x,y) \supset Q(x,y)) \vdash (\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}\ \text{cut}$$

where φ_l is

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(z,a)^* \vdash P(z,a)}{\vdash \underline{\neg P(z,a)^*}, P(z,a)}\ \neg:\mathrm{r}}{\vdash \underline{\neg P(z,a) \vee Q(z,a)^*}, P(z,a)}\ \vee:\mathrm{r}_1 \quad \dfrac{Q(z,a) \vdash Q(z,a)^*}{Q(z,a) \vdash \underline{\neg P(z,a) \vee Q(z,a)^*}}\ \vee:\mathrm{r}_2}{\underline{P(z,a) \supset Q(z,a)} \vdash \neg P(z,a) \vee Q(z,a)^*}\ \supset:\mathrm{l}}{\underline{(\forall y)(P(z,y) \supset Q(z,y))} \vdash \neg P(z,a) \vee Q(z,a)^*}\ \forall:\mathrm{l}}{\underline{(\forall x)(\forall y)(P(x,y) \supset Q(x,y))} \vdash \neg P(z,a) \vee Q(z,a)^*}\ \forall:\mathrm{l}}{(\forall x)(\forall y)(P(x,y) \supset Q(x,y)) \vdash \underline{(\exists y)(\neg P(z,y) \vee Q(z,y))^*}}\ \exists:\mathrm{r}}{(\forall x)(\forall y)(P(x,y) \supset Q(x,y)) \vdash \underline{(\forall x)(\exists y)(\neg P(x,y) \vee Q(x,y))^*}}\ \forall:\mathrm{r}$$

and φ_r is

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(b,v) \vdash P(b,v)^*}{\underline{\neg P(b,v)^*}, P(b,v) \vdash}\ \neg:\mathrm{l}}{\neg P(b,v)^* \vdash \underline{\neg P(b,v)}}\ \neg:\mathrm{r} \quad \dfrac{Q(b,v)^* \vdash Q(b,v)}{\underline{\neg Q(b,v)}, Q(b,v)^* \vdash}\ \neg:\mathrm{l}}{\neg Q(b,v), \underline{\neg P(b,v) \vee Q(b,v)^*} \vdash \neg P(b,v)}\ \vee:\mathrm{l}'}{\neg P(b,v) \vee Q(b,v)^* \vdash \underline{\neg Q(b,v) \supset \neg P(b,v)}}\ \supset:\mathrm{r}}{\neg P(b,v) \vee Q(b,v)^* \vdash \underline{(\exists y)(\neg Q(b,y) \supset \neg P(b,y))}}\ \exists:\mathrm{r}}{\neg P(b,v) \vee Q(b,v)^* \vdash \underline{(\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}}\ \exists:\mathrm{r}}{\underline{(\exists y)(\neg P(b,y) \vee Q(b,y))^*} \vdash (\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}\ \exists:\mathrm{l}}{\underline{(\forall x)(\exists y)(\neg P(x,y) \vee Q(x,y))^*} \vdash (\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}\ \forall:\mathrm{l}$$

The extraction of the characteristic clause term happens top down starting with those parts of the initial sequents that are marked as ancestors of cut formulas which are now interpreted as sets. At every occurrence of a binary rule the two clause terms resulting from the premises are connected by a binary operator. Depending whether the auxiliary formulas of the inference were ancestors of cut formulas or not the operator will either be $\oplus$ or $\otimes$. All unary inference rules have no influence on the clause term and hence it remains unchanged.

For the example above we obtain the following characteristic clause term

$$\Theta(\varphi) = ((\{P(z,a) \vdash\} \otimes \{\vdash Q(z,a)\}) \oplus (\{\vdash P(b,v)\} \oplus \{Q(b,v) \vdash\}))$$

which characterizes those parts of the axiom sequents which have been used to derive the cut formula (on both sides).

The operator $\oplus$ of the clause term is interpreted as union and the operator $\otimes$ as merge, i.e. the antecedent and consequent parts of different sequents are exchanged such that only one part is exchanged

at once.

Hence by evaluation of $\Theta(\varphi)$ for the characteristic clause set $|\Theta(\varphi)|$ of φ we obtain

$$|\Theta(\varphi)| = \{P(z,a) \vdash Q(z,a), \quad (C_2)$$
$$\vdash P(b,v), \quad (C_1)$$
$$Q(b,v) \vdash\}. \quad (C_3)$$

The characteristic clause set of an **LK** derivation is always unsatisfiable. Therefore one can always find a resolution refutation of the characteristic clause set.

In particular, we define a resolution refutation δ of $|\Theta(\varphi)|$:

$$\dfrac{Q(b,v) \vdash \qquad \dfrac{\vdash P(b,v) \quad P(z,a) \vdash Q(z,a)}{\vdash Q(b,a)}}{\vdash}$$

and a corresponding ground refutation γ of δ, i. e. $\gamma = \delta\sigma$:

$$\dfrac{Q(b,a) \vdash \qquad \dfrac{\vdash P(b,a) \quad P(b,a) \vdash Q(b,a)}{\vdash Q(b,a)}}{\vdash}$$

with the ground substitution $\sigma = \{v \mapsto a, z \mapsto b\}$.

Now we have to reduce φ to projections of the clauses used as initial clauses in the resolution refutation of $|\Theta(\varphi)|$. This projections may be understood as projection schemes of the clauses in question modulo a corresponding ground substitution.

Again, we start at the initial sequents (without those parts marked as ancestors of cut formulas and not necessary for the creation of the clause in question) and apply all inference rules not operating on ancestors of cut formulas until all such binary rules have been applied and at least one formula also occurring in the end sequent has been composed.

The projection of φ to the clause C_1 is:

$$\varphi(C_1) = \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(b,v) \vdash P(b,v)}{\vdash P(b,v), \neg P(b,v)}\,\neg : r}{\neg Q(b,v) \vdash P(b,v), \neg P(b,v)}\,w : l}{\vdash \neg Q(b,v) \supset \neg P(b,v), P(b,v)}\,\supset : r}{\vdash (\exists y)(\neg Q(b,y) \supset \neg P(b,y)), P(b,v)}\,\exists : r}{\vdash (\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y)), P(b,v)}\,\exists : r$$

and the corresponding ground projection $\chi_1 = \varphi(C_1)\sigma$.

The projection of φ to the clause C_2 is:

$\varphi(C_2) =$

$$\dfrac{\dfrac{\dfrac{P(z,a) \vdash P(z,a) \quad Q(z,a) \vdash Q(z,a)}{P(z,a) \supset Q(z,a), P(z,a) \vdash Q(z,a)} \supset : l}{(\forall y)(P(z,y) \supset Q(z,y)), P(z,a) \vdash Q(z,a)} \forall : l}{(\forall x)(\forall y)(P(x,y) \supset Q(x,y)), P(z,a) \vdash Q(z,a)} \forall : l$$

with ground projection $\chi_2 = \varphi(C_2)\sigma$.

And finally the projection of φ to the clause C_3:

$\varphi(C_3) =$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Q(b,v) \vdash Q(b,v)}{\neg Q(b,v) Q(b,v) \vdash} \neg : l}{\neg Q(b,v) Q(b,v) \vdash \neg P(b,v)} w : r}{Q(b,v) \vdash \neg Q(b,v) \supset \neg P(b,v)} \supset : r}{Q(b,v) \vdash (\exists y)(\neg Q(b,y) \supset \neg P(b,y))} \exists : r}{Q(b,v) \vdash (\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))} \exists : r$$

with ground projection $\chi_3 = \varphi(C_3)\sigma$.

Finally the ground projections can be composed to a cut-free proof of φ, i.e. a proof of φ containing only atomic cuts, using its resolution refutation as a skeleton.

$$\dfrac{\dfrac{\begin{matrix}(\chi_1) \\ \vdash Y, P(b,a)\end{matrix} \quad \begin{matrix}(\chi_2) \\ P(b,a), X \vdash Q(b,a)\end{matrix}}{X \vdash Y, Q(b,a)} \text{ cut} \quad \begin{matrix}(\chi_3) \\ Q(b,a) \vdash Y\end{matrix}}{X \vdash Y} \text{ cut}$$

where $X = (\forall x)(\forall y)(P(x,y) \supset Q(x,y))$ and $Y = (\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))$.

The schematic CERES-method generalizes the approach described above to infinite sequences of proofs defined in form of recursion (see [8]). A recursive definition of the sequence of characteristic clause sets can be algorithmically extracted; the same holds for the sequences of projections. The most difficult part is the specification of an infinite sequence of resolution refutations and the corresponding unifiers; a language for schematic resolution refutation was defined in [8], but it is not known if every schematic clause set derived from a proof in the $\mathbf{LKS}_\varepsilon$ calculus has a refutation which is expressible by this language. Even more so, if there exists a refutation of a schematic clause set expressible by the language of [8], finding that refutation in particular is an undecidable problem.

3 Analysis of the Schematic Tape Proof

3.1 Formalization of the Tape proof

In this section we will formalize a proof of the following statement:

Statement 1 (Tape Statement (introduced in [2]))**.** *Given a total function $f : \mathbb{N} \to \mathbb{N}_n$, where $n \in \mathbb{N}$, there exists $i, j \in \mathbb{N}$ such that $i < j$ and $f(i) = f(j)$.*

In particular we will use the following schema of lemmata to prove the tape statement:

Lemma 1 (Infinity Lemma)**.** *Given a total function $f : \mathbb{N} \to \mathbb{N}_{n+1}$ then either for all $x \in \mathbb{N}$ there exist a $y \in \mathbb{N}$ such that $x \leq y$ and $f(y) = i$ where $i \in \mathbb{N}_n$, or for all $x \in \mathbb{N}$ there exist a $y \in \mathbb{N}$ such that $x \leq y$ and $f(y) = n+1$.*

Essentially, the proof of the tape statement is proved by induction on the free variable n in the infinity lemma. We formalize this proof in the $\mathbf{LKS}_\varepsilon$ calculus [8]. The only rewrite rules, i.e. the set of rules ε, needed for the defined predicate symbols will be those for *iterated conjunction* and *iterated disjunction*:

$$\left(\bigwedge_{i=0}^{n} \varphi\,[i \setminus x]\right) \wedge \varphi\,[(n+1) \setminus x] \Rightarrow \bigwedge_{i=0}^{n+1} \varphi\,[i \setminus x]$$

$$\left(\bigvee_{i=0}^{n} \varphi\,[i \setminus x]\right) \vee \varphi\,[(n+1) \setminus x] \Rightarrow \bigvee_{i=0}^{n+1} \varphi\,[i \setminus x]$$

When the upper bound of the interval is less than 0 the iterated conjunction is equivalent to $\top$ and the iterated disjunction is equivalent to $\bot$.

The only other addition to the standard **LK** calculus [15] is the addition of *proof links* at the leaves of the proofs, but before introducing proof links we provide a few additional constructions necessary for formalization of schematic proofs. A schematic proof written in the $\mathbf{LKS}_\varepsilon$ calculus is written as a *proof schema*, a sequence of pairs of proofs. The first proof in the set is the root of the schema. We will refer to these pairs as *proof schema pairs*. Each proof schema pair has a *proof symbol* representing it, and a base-case and step-case end sequent indexed by a term in the numeric sort. In our case the proof schema is $\langle(\omega(0), \omega(n+1)), (\varphi(0), \varphi(n+1))\rangle$. The left to right ordering of the pairs in our proof schema implies that the proof schema pair for ω can contain a call to the pair for φ or a call to itself, and the pair for φ can only call itself. Proofs in a proof schema pair cannot make calls to proofs further to the left than they are located. The end sequents is as follows ($\mathbf{es}(\cdot)$ means the end sequent of given proof symbol):

$$\begin{gathered}
\mathbf{es}(\omega(0)) = \forall x f(x) \sim 0 \vdash \exists p \exists q (p < q \wedge f(p) \sim f(q)) \\
\mathbf{es}(\omega(n+1)) = \forall x \textstyle\bigvee_{i=0}^{n+1} f(x) \sim i \vdash \exists p \exists q (p < q \wedge f(p) \sim f(q)) \\
\mathbf{es}(\varphi(0)) = \forall x \exists y (x \leq y \wedge f(y) \sim 0) \vdash \exists p \exists q (p < q \wedge f(p) \sim f(q)) \\
\mathbf{es}(\varphi(n+1)) = \forall x \exists y (x \leq y \wedge \textstyle\bigvee_{i=0}^{n+1} f(y) \sim i) \vdash \exists p \exists q (p < q \wedge f(p) \sim f(q)).
\end{gathered}$$

We use the symbol $\sim$ to represent equality over the numeric sort. In the case of proof $\varphi(n+1)$ one of the leaves has a call to the case of $\varphi(n)$. Proof links are used to represent this call to a previous proof and are written as follows:

$$\frac{\varphi(n)}{\forall x \exists y (x \leq y \wedge \textstyle\bigvee_{i=0}^{n} f(y) \sim i) \vdash \exists p \exists q (p < q \wedge f(p) \sim f(q))}$$

In general, proof links can also take arguments from the individual sort, but this is not needed in our case. When we instantiate a proof schema for a given value of the free parameter, we replace the proof links with instances of the proof indicated by the proof symbol in the proof link and the numeric term.

We will now outline the proof, skipping many of the trivial sequent rules, and demarcating the cut ancestors with *. Also, instead of using $=$ to represent equality in the formal proof we will use $\sim$ being that all equality is occurring over the numeric sort. f should be understood as a function mapping the members of the individual sort to the numeric sort. By $s(\cdot)$ we mean successor function over the individual sort.

3.1.1 Proof Symbol ω Base-case

$$\dfrac{\dfrac{\vdash \alpha \leq \alpha^* \qquad \begin{array}{c} f(\alpha) \sim 0 \vdash \\ f(\alpha) \sim 0^* \end{array}}{\vdots}\wedge : r \qquad \dfrac{\dfrac{\begin{array}{c} s(\beta) \leq \alpha^* \vdash \\ \beta < \alpha \end{array} \qquad \begin{array}{c} f(\beta) \sim 0^*, f(\alpha) \sim 0^* \vdash \\ f(\beta) \sim f(\alpha) \end{array}}{\vdots}\wedge : r}{\begin{array}{c} \forall x f(x) \sim 0 \vdash \\ \forall x \exists y (x \leq y \wedge f(y) \sim 0)^* \end{array} \qquad \begin{array}{c} \forall x \exists y (x \leq y \wedge f(y) \sim 0)^* \vdash \\ \exists p \exists q (p < q \wedge f(p) \sim f(q)) \end{array}}cut$$
$$\forall x f(x) \sim 0 \vdash$$
$$\exists p \exists q (p < q \wedge f(p) \sim f(q))$$

3.1.2 Proof Symbol ω Step-case

$$\dfrac{\dfrac{\varphi(n+1)}{\forall x \exists y (x \leq y \wedge \bigvee_{i=0}^{n+1} f(y) \sim i)^* \vdash} \qquad \dfrac{\dfrac{\vdash \alpha \leq \alpha^* \qquad \begin{array}{c} \bigvee_{i=0}^{n+1} f(\alpha) \sim i \vdash \\ \bigvee_{i=0}^{n+1} f(\alpha) \sim i^* \end{array}}{\vdots}\wedge : r}{\begin{array}{c} \forall x \bigvee_{i=0}^{n+1} f(x) \sim i \vdash \\ \forall x \exists y (x \leq y \wedge \bigvee_{i=0}^{n+1} f(y) \sim i)^* \end{array}}cut$$
$$\exists p \exists q (p < q \wedge f(p) \sim f(q))$$
$$\forall x \bigvee_{i=0}^{n+1} f(x) \sim i \vdash$$
$$\exists p \exists q (p < q \wedge f(p) \sim f(q))$$

3.1.3 Proof Symbol φ base-case

$$\dfrac{\dfrac{\begin{array}{c} s(\beta) \leq \alpha^* \vdash \\ \beta < \alpha \end{array} \qquad \begin{array}{c} f(\beta) \sim 0^*, f(\alpha) \sim 0^* \vdash \\ f(\beta) \sim f(\alpha) \end{array}}{\vdots}\wedge : r}{\begin{array}{c} \forall x \exists y (x \leq y \wedge f(y) \sim 0)^* \vdash \\ \exists p \exists q (p < q \wedge f(p) \sim f(q)) \end{array}}$$

3.1.4 Proof Symbol φ step-case

In this proof we will mark *cut-configuration ancestors*[2] with **. These are cut-ancestors that passed through proof links.

[2]The point of the configurations is to track the cut-status of formulae that pass through proof links (whether the formulae are ancestors of cuts or not). A configuration is a set of formula occurrences from the end-sequent of a given proof[8].

$$\dfrac{\dfrac{\dfrac{max(\alpha,\beta)\leq\gamma^{**}\vdash \quad \bigvee_{i=0}^{n+1} f(\gamma)\sim i^{**},\ \vdash f(\gamma)\sim n+1^{*}\ \bigvee_{i=0}^{n} f(\gamma)\sim i^{*}}{\alpha\leq\gamma^{*}}}{max(\alpha,\beta)\leq\gamma^{**},\bigvee_{i=0}^{n+1} f(\gamma)\sim i^{**},\ \vdash f(\gamma)\sim n+1^{*}\ \alpha\leq\gamma\wedge\bigvee_{i=0}^{n} f(\gamma)\sim i^{*}}\wedge:r \qquad max(\alpha,\beta)\leq\gamma^{**}\vdash\ \beta\leq\gamma^{*}}{\vdots\ \forall x\exists y(x\leq y\wedge\bigvee_{i=0}^{n+1} f(y)\sim i)^{**}\vdash\ \forall x\exists y(x\leq y\wedge\bigvee_{i=0}^{n} f(y)\sim i)^{*},\ \forall x\exists y(x\leq y\wedge f(y)=n+1)^{*}\ \vdots}\wedge:r$$

$$\dfrac{\dfrac{\vdots\ \forall x\exists y(x\leq y\wedge\bigvee_{i=0}^{n+1} f(y)\sim i)^{**}\vdash\ \forall x\exists y(x\leq y\wedge\bigvee_{i=0}^{n} f(y)\sim i)^{*},\ \forall x\exists y(x\leq y\wedge f(y)=n+1)^{*} \qquad \dfrac{\varphi(n)}{\forall x\exists y(x\leq y\wedge\bigvee_{i=0}^{n} f(y)\sim i)^{*}\vdash\ \exists p\exists q(p<q\wedge f(p)\sim f(q))}}{\forall x\exists y(x\leq y\wedge\bigvee_{i=0}^{n+1} f(y)\sim i)^{**}\vdash\ \exists p\exists q(p<q\wedge f(p)\sim f(q)),\ \forall x\exists y(x\leq y\wedge f(y)\sim n+1)^{*}\ \vdots}cut$$

$$\dfrac{\dfrac{\vdots\ \forall x\exists y(x\leq y\wedge\bigvee_{i=0}^{n+1} f(y)\sim i)^{**}\vdash\ \exists p\exists q(p<q\wedge f(p)\sim f(q)),\ \forall x\exists y(x\leq y\wedge f(y)\sim n+1)^{*} \qquad \dfrac{\dfrac{s(\beta)\leq\alpha^{*}\vdash}{\beta<\alpha} \quad \dfrac{f(\beta)\sim(n+1)^{*},f(\alpha)\sim(n+1)^{*}\vdash}{f(\beta)\sim f(\alpha)}\wedge:r}{\vdots\ \forall x\exists y(x\leq y\wedge f(y)\sim n+1)^{*}\vdash\ \exists p\exists q(p<q\wedge f(p)\sim f(q))}}{\forall x\exists y(x\leq y\wedge\bigvee_{i=0}^{n+1} f(y)\sim i)^{**}\vdash\ \exists p\exists q(p<q\wedge f(p)\sim f(q)),\ \exists p\exists q(p<q\wedge f(p)\sim f(q))}cut$$

$$\dfrac{\forall x\exists y(x\leq y\wedge\bigvee_{i=0}^{n+1} f(y)\sim i)^{**}\vdash\ \exists p\exists q(p<q\wedge f(p)\sim f(q)),\ \exists p\exists q(p<q\wedge f(p)\sim f(q))}{\forall x\exists y(x\leq y\wedge\bigvee_{i=0}^{n+1} f(y)\sim i)^{**}\vdash\ \exists p\exists q(p<q\wedge f(p)\sim f(q))}c:r$$

Extracting the clause sets from the base cases of $\omega(0)$ and $\varphi(0)$, the following two sets result:

$$CS(\omega,0,\emptyset)\equiv((((s(\beta)\leq\alpha\vdash)\otimes(f(\alpha)\sim 0,f(\beta)\sim 0\vdash))\oplus\vdash\alpha\leq\alpha)\oplus\vdash f(\alpha)\sim 0)$$

$$CS(\varphi,0,\emptyset)\equiv(s(\beta)\leq\alpha\vdash)\otimes(f(\alpha)\sim 0,f(\beta)\sim 0\vdash)$$

By $CS(\omega,0,\emptyset)$ we are referring to the clause set for proof ω given the value 0 from the numeric set and configuration $\emptyset$. Extracting the clause set from the step-case of φ will require us to use a single configuration, namely, $\Omega_{n+1}=\left\{\forall x\exists y(x\leq y\wedge\bigvee_{i=0}^{n+1} f(y)\sim i)^{**}\vdash\right\}$. This is the cut formula passed through the proof links.

$$CS(\omega,n+1,\emptyset)\equiv(((\vdash\alpha\leq\alpha)\oplus(\vdash\bigvee_{i=0}^{n+1} f(\alpha)\sim i))\oplus CS(\varphi,n+1,\Omega_{n+1}))$$

$$CS(\varphi, n+1, \Omega_{n+1}) \equiv (((((s(\beta) \leq \alpha \vdash) \otimes (f(\alpha) \sim n+1, f(\beta) \sim n+1 \vdash)) \oplus (max(\alpha,\beta) \leq \gamma \vdash \alpha \leq \gamma)) \oplus$$

$$(max(\alpha,\beta) \leq \gamma \vdash \beta \leq \gamma)) \oplus CS(\varphi, n, \Omega_n))$$

When we simplify this representation and remove redundancy, we get the following clause set $C(n)$:

$$\begin{array}{cc}
(C1) & \vdash \alpha \leq \alpha \\
(C2) & max(\alpha,\beta) \leq \gamma \vdash \alpha \leq \gamma \\
(C3) & max(\alpha,\beta) \leq \gamma \vdash \beta \leq \gamma \\
(C4_0) & f(\beta) \sim 0, f(\alpha) \sim 0, s(\beta) \leq \alpha \vdash \\
\vdots & \vdots \\
(C4_n) & f(\beta) \sim n, f(\alpha) \sim n, s(\beta) \leq \alpha \vdash \\
(C5) & \vdash f(\alpha) \sim 0, \cdots, f(\alpha) \sim n
\end{array}$$

The clause set $C(n)$ is the set which we will prove unsatisfiable for all instantiations of the free parameter n. Being that the language for resolution refutations presented in the original paper on schematic CERES [8] does not have enough expressive power, we will present the refutation using a mathematical meta-language.

3.2 Schematic Resolution Refutation of $C(n)$

Definition 1. *Let the* variable symbol set $\mathscr{V}$ *denote a countably infinite set of variable symbols. An* indexed variable x_i *is a first-order variable with* $x \in \mathscr{V}$ *and* $i \in \mathbb{N}$.

Definition 2. *The primitively recursively defined term* $m_n(k, x_0, \cdots, x_n)$ *with arity* $n+1$ *for* $k, n \in \mathbb{N}$, $x \in \mathscr{V}$ *is defined as follows:*

$$When\ n < k+1, \quad m_n(k+1, x_0, \cdots, x_n) \Rightarrow m_n(k, x_0, \cdots, x_n) \tag{2a}$$

$$When\ k+1 \leq n, \qquad \begin{array}{c} m_n(k+1, x_0, \cdots, x_n) \Rightarrow \\ max(m_n(k, x_0, \cdots, x_n), s(x_{k+1})) \end{array} \tag{2b}$$

$$m_n(0, x_0, \cdots, x_n) \Rightarrow max(s(x_0), s(x_0)) \tag{2c}$$

We will use the abbreviation $\bar{x}_n$ *for the list of parameters* $x_0, \cdots, x_n$. *Also, for simplicity we will always write the term* $m_n(k, x_0, \cdots, x_n)$ *as* $m_n(k, \bar{x}_n)$ *where* $k \leq n$, *being that the function skips evaluation for values higher than n.*

Example 1. *The term* $m_2(2, x_0, x_1, x_2)$ *when unrolled will be the following:*

$$max(max(max(s(x_0), s(x_0)), s(x_1), s(x_2))$$

This definition of a nested max term will be integral to the refutation of the clause set and results in the following lemma about clauses derivable from $C(n)$.

Lemma 2. *Given* $0 \leq k \leq n$, *the clause* $\vdash m_n(k, \bar{x}_n) \leq m_n(n, \bar{x}_n)$ *is derivable from (C1),(C2),and (C3).*

Proof. We prove this lemma by induction on the difference between n and k. When $n = k$, the only possibility is $\vdash m_n(n,\bar{x}_n) \leq m_n(n,\bar{x}_n)$, an instance of $(C1)$. Assuming for all differences $u' \leq u$ the lemma holds, we now show for $u+1$. This leaves two possibilities of which we will prove one and leave the other to the reader, namely, $u+1 = n-(k-1)$ for $k > 1$. This implies deriving the clause $\vdash m_n(k-1,\bar{x}_n) \leq m_n(n,\bar{x}_n)$.

1. $\vdash max(\alpha,\beta) \leq \gamma \rightarrow \alpha \leq \gamma$	(C2)
2. $m_n(k,\bar{x}_n) \leq m_n(n,\bar{x}_n) \vdash m_n(k-1,\bar{x}_n) \leq m_n(n,\bar{x}_n)$	By definition of m_n and substitution into 1.
3. $\vdash m_n(k,\bar{x}_n) \leq m_n(n,\bar{x}_n)$	Induction hypothesis
4.$\vdash m_n(k-1,\bar{x}_n) \leq m_n(n,\bar{x}_n)$	resolve 2 and 3.

□

Lemma 3. *Given $0 \leq k \leq n$, the clause $\vdash s(x_k) \leq m_n(n,\bar{x}_n)$ is derivable from (C1),(C2),and (C3).*

Proof. We know that for every n and k, $\vdash m_n(k,\bar{x}_n) \leq m_n(n,\bar{x}_n)$ is derivable by Lem. 2. Thus the following derivations show that $\vdash s(x_k) \leq m_n(n,\bar{x}_n)$ is derivable:

1. $\vdash max(\alpha,\beta) \leq \gamma \rightarrow \beta \leq \gamma$	(C3)
2. $m_n(k,\bar{x}_n) \leq m_n(n,\bar{x}_n) \vdash s(x_k) \leq m_n(n,\bar{x}_n)$	By definition of m_n and substitution into 1.
3. $\vdash m_n(k,\bar{x}_n) \leq m_n(n,\bar{x}_n)$	Lem. 2
4.$\vdash s(x_k) \leq m_n(n,\bar{x}_n)$	Resolve 2 and 3.

□

Lemma 4. *Given $0 \leq i \leq n$, the clause $f(m_n(n,\bar{x}_n)) \sim i, f(x_i) \sim i \vdash$ is derivable from (C1),(C2),(C3),and ($C4_i$).*

Proof. By Lem. 3 $\vdash s(x_i) \leq m_n(n,\bar{x}_n)$ is derivable, thus the following holds:

1. $\vdash s(x_i) \leq m_n(n,\bar{x}_n)$	Lem. 3
2. $f(x) \sim i, f(y) \sim i, s(x) \leq y \vdash$	($C4_i$).
3. $f(x_i) \sim i, f(m_n(n,\bar{x}_n)) \sim i, s(x_i) \leq m_n(n,\bar{x}_n) \vdash$	Substitution into 2.
4.$f(m_n(n,\bar{x}_n)) \sim i, f(x_i) \sim i \vdash$	Resolve 1 and 3.

□ □

So far the steps taken in the refutation of the clause set $C(n)$ have been straightforward and have not required much additional machinery to derive. However, in the next lemma we need to add bijective functions to the numeric sort to show the derivability of certain clauses. The problem we need to get around is that every permutation of the numbers from 0 to n will be needed to refute the clause set. Thus, we cannot directly prove the properties of the refutation using a simple linear induction. We replace the numbers in the numeric sort with the bijective function in order to simplify the problem enough to prove the derivability of some clauses without using a complex ordering. However, even this simplification was not enough for all the clauses needed in the refutation and to derive $\vdash$ we still need to construct a special ordering.

Definition 3. *Given $0 \leq n$, $-1 \leq k \leq j \leq n$, and a bijective function $o : \mathbb{N}_n \rightarrow \mathbb{N}_n$ we define the following formulae:*

$$c_o(k,j,n) = \bigwedge_{i=0}^{k} f(x_{o(i)}) \sim o(i) \vdash \bigvee_{i=k+1}^{j} f(m_n(n,\bar{x}_n)) \sim o(i).$$

The formulae $c_o(-1,-1,n) \equiv \vdash$ for all values of n.

Lemma 5. *Given $0 \leq n$, $-1 \leq k \leq n$ and for all bijective functions $o : \mathbb{N}_n \to \mathbb{N}_n$, the formula $c_o(k,n,n)$ is derivable from C(n).*

Proof. We prove this lemma using an induction on k and a case distinction on n. When $n = 0$ there are two possible values for k, $k = 0$ or $k = -1$. When $k = -1$ the clause is an instance of (C5). When $k = 0$ we have the following derivation (remember that the bijective function must be $o(0) = 0$):

1. $f(m_n(n,\bar{x}_n)) \sim 0, f(x_0) \sim 0 \vdash$	Lemma 4
2. $\vdash f(x) \sim 0$	(C5).
3. $\vdash f(m_n(n,\bar{x}_n)) \sim 0$	Substitution into 2.
4. $f(x_0) \sim 0 \vdash$	Resolution 1 and 3.

When $n > 0$ and $k = -1$ we again trivially have (C5). Now we assume that for all $w \leq k$ for $n > 0$ the theorem holds, we then proceed to prove the theorem holds for $k+1$. We assume that $k < n$. The following derivation will suffice:

1. $\bigwedge_{i=0}^{k} f(x_{o(i)}) \sim o(i) \vdash \bigvee_{i=k+1}^{n} f(m_n(n,\bar{x}_n)) \sim o(i)$	Induction hypothesis
2. $(f(m_n(n,\bar{x}_n)) \sim o(k+1) \wedge f(x_{o(k+1)}) \sim o(k+1)) \vdash$	Lem. 4
3. $\bigwedge_{i=0}^{k+1} f(x_{o(i)}) \sim o(i) \vdash \bigvee_{i=k+2}^{n} f(m_n(n,\bar{x}_n)) \sim o(i)$	Resolve 1 and 2.

□

Definition 4. *Given $0 \leq n$ we define the ordering relation $\lessdot_n$ over $A_n = \{(i,j) | i \leq j \wedge 0 \leq i,j \leq n \wedge i,j \in \mathbb{N}\}$ s.t. for $(i,j),(l,k) \in A_n$, $(i,j) \lessdot_n (l,k)$ iff $i,k,l \leq n$, $j < n$, $l \leq i$, $k \leq j$, and $i = l \leftrightarrow j \neq k$ and $j = k \leftrightarrow i \neq l$.*

The ordering defined above is essentially the resolution refutation. It is a complex and strange ordering, however, it allows for a simple and straight forward schematically definable refutation.

Lemma 6. *The ordering $\lessdot_n$ over A_n for $0 \leq n$ is a complete well ordering.*

Proof. Every chain has a greatest lower bound, namely, one of the members of A_n, (i,n) where $0 \leq i \leq n$, and it is transitive, anti-reflexive, and anti-symmetric. □

The clauses proved derivable by Lem. 5 can be paired with members of A_n as follows, $c_o(k,n,n)$ is paired with (k,n). Thus, each $c_o(k,n,n)$ is essentially the greatest lower bound of some chain in the ordering $\lessdot_n$ over A_n.

Lemma 7. *Given $0 \leq k \leq j \leq n$, for all bijective functions $o : \mathbb{N}_n \to \mathbb{N}_n$ the clause $c_o(k,j,n)$ is derivable from C(n).*

Proof. We will prove this lemma by induction over A_n. The base cases are the clauses $c_o(k,n,n)$ from Lem. 5. Now let us assume that the lemma holds for all clauses $c_o(k,i,n)$ pairs such that, $0 \leq k \leq j < i \leq n$ and for all clauses $c_o(w,j,n)$ such that $0 \leq k < w \leq j \leq n$, then we want to show that the lemma holds for the clause $c_o(k,j,n)$. The following derivation provides proof:

1. $\bigwedge_{i=0}^{k} f(x_{o(i)}) \sim o(i) \vdash \bigvee_{i=k+1}^{j+1} f(m_n(n,\bar{x}_n)) \sim o(i)$	Induction hypothesis $c_o(k,j+1,n)$
2. $\bigwedge_{i=0}^{k+1} f(x_{o'(i)}) \sim o'(i) \vdash$	Induction hypothesis $c_{o'}(k+1,k+1,n)$
3. $\bigwedge_{i=0}^{k} f(x_{o(i)}) \sim o(i) \vdash \bigvee_{i=k+1}^{j} f(m_n(n,\bar{x}_n)) \sim o(i)$	Resolve 1,2 $\{x_{o'(k+1)} \leftarrow m_n(n,\bar{x}_n)\}$

Also, $o'(e) = o(e)$ for $0 \leq e \leq k$, $o'(k+1) = o(j+1)$ and $o(k+1) = o'(e)$ for $k+1 < e \leq n$. □

Theorem 1. *Given $n \geq 0$, $C(n)$ derives $\vdash$.*

Proof. By Lem. 7, The clauses $f(x) \sim 0 \vdash$, $\cdots$, $f(x) \sim n \vdash$ are derivable. Thus, we can prove the statement by induction on the instantiation of the clause set. When $n = 0$, the clause (C5) is $\vdash f(x) \sim 0$ which resolves with $f(x) \sim 0 \vdash$ to derive $\vdash$. Assuming that for all $n' \leq n$ the theorem holds we now show that it holds for $n+1$. The clause (C5) from the clause set $C(n+1)$ is the clause (C5) from the clause set $C(n)$ with the addition of a positive instance of $\vdash f(\alpha) \sim (n+1)$. Thus, by the induction hypothesis we can derive the clause $\vdash f(\alpha) \sim (n+1)$. By Lem. 7 we can derive $f(x) \sim (n+1) \vdash$, and thus, resolving the two derived clauses results in $\vdash$.

□

It is still an open problem as to whether this is the minimal schematic refutation of this clause set; however, evidence points towards this being the case. One way to measure the complexity of the refutation is to count how many times schematic length clauses are used. In this clause set we only have one schematic length clause (a clause whose size is dependent on the free parameter), namely, (C5). By Thm. 1 we see that the last step requires one instance of (C5) plus n+1 times the number of instances needed to derive $f(x) \sim o(i) \vdash$. This pattern occurs throughout the proof of Lem. 7. Thus, we can consider the recurrence $f(n+1) = 1 + (n+1) \cdot f(n)$. This recurrence is roughly equivalent to the function $e \cdot n!$ implying that the number of schematic clauses needed for this refutation is enormous and the refutation is very redundant. This also provides evidence as to the difficulty of developing a new language for the resolution refutation calculus.

As mentioned, the schematic ACNF (see [8]) for this proof uses many instances of the schematic length clause, is complex, and most likely obfuscates the information we would like to extract during proof analysis, the weak quantifier instantiations. Part of the problem with the resolution refutation, at least in its current form, is that a global unifier for the schematic proof is very hard to construct. But, we do have the local unifiers, which are pretty much the same unifier repeated at every step in Lem. 7. Unlike in the case of ANCF construction where the abundance of some clauses gets in the way, the redundancy of the local unifiers helps us construct a schematic Herbrand sequent. In the next section we will show how a schematic Herbrand sequent (for this specific case) can be constructed. No general method exist as of yet. Though, we will not construct a minimal Herbrand sequent, we show that patterns in the resolution refutation lead to the extraction of the finite pigeon-hole principle.

4 Herbrand Sequent Extraction

Traditionally, the projections constructed from a formalized proof for the CERES method are built algorithmically from the structure of the said proof. This holds true for the schematic CERES method as well. However, being that we have not been able to construct a global unifier, if we want to get information about which instantiation of the weak quantifiers are needed, we need to know how the local unifiers change the terms at the leaves of the derivation (our proof does not have ground axioms).

We introduce *quasi-projections* which are the same as projections, except we drop the weak quantifier rules from the projections and we do not weaken in the end-sequent formulae, which are not proven from the axioms used in the given projection. This concept is still in its infancy and will probably need more work to generalize it to arbitrary $\mathbf{LKS}_\varepsilon$ projections. This results in the formulae which are ancestors of the cut sharing variables with the ancestors of the end-sequent. Thus as we apply the local unifiers used in the resolution refutation to the clauses extended with the end-sequent formulae, we can gather the instantiations for the weak quantifiers. This method does not have much amount to much concerning the instantiations up to Lem. 5; however, for the more complex induction found in Lem. 7 a pattern emerges. The only quasi projections we need to construct are for clauses ($C4_i$) and (C5):

4.0.1 Quasi-Projection for (C4$_i$)

$$\frac{s(\alpha) \leq \beta \vdash \alpha < \beta \qquad \begin{array}{c} f(\alpha) \sim i, f(\beta) \sim i \vdash \\ f(\alpha) \sim f(\beta) \end{array}}{\begin{array}{c} f(\alpha) \sim i, f(\beta) \sim i, s(\alpha) \leq \beta \vdash \\ \alpha < \beta \wedge f(\alpha) \sim f(\beta) \end{array}} \wedge : r$$

4.0.2 Quasi-Projection for (C5)

$$\bigvee_{j=0}^{n} f(\beta) \sim j \vdash \bigvee_{j=0}^{n} f(\beta) \sim j$$

One more problem to deal with concerning the construction of the Herbrand sequent is given the fact that a factorial number of schematic length clauses are necessary for the refutation, the number of local unifiers will be even more numerous, leaving us again with a plethora of mostly useless information (a lot of repeated instances). We are able to alleviate this issue by introducing equivalence classes based on the nesting depth of the iterated max function of definition 2. The following definitions cover the equivalence classes.

Definition 5. *We define $dom(\sigma_x)$ as the domain of σ_x. Let σ_x where $x \in \mathcal{V}$ be a substitution such that $x \notin dom(\sigma_x)$ and in the range σ_x the only variable symbol is x.*

Definition 6. *Let $\mathcal{N}^k$ for $k \in \mathbb{N}$ and $x \in \mathcal{V}$ be the class of all equivalences classes $\overline{\mathbf{n}}_{(x,k)}$ of the following form:*

$$\overline{\mathbf{0}}_{(x,k)} \equiv \{x_i | i \in [0,k]\}$$

$$\overline{\mathbf{n}}_{(x,k)} \equiv \left\{ m_k(k, \bar{y}_k)\sigma_x \middle| \sigma_x = \left\{ y_0 \leftarrow w_0, \cdots, y_{i-1} \leftarrow w_{i-1}, y_i \leftarrow w_i, y_{i+1} \leftarrow w_{i+1}, \cdots, y_k \leftarrow w_k \right\} \right.$$
$$\left. y \in \mathcal{V}, i \in [0,k], \left(\bigwedge_{j \in [0,\cdots i-1, i+1, \cdots k]} \bigvee_{v \in \left[\overline{\mathbf{0}}_{(x,k)}, \cdots, \overline{\mathbf{n-1}}_{(x,k)}\right]} w_j \in v \right), w_i \in \overline{\mathbf{n-1}}_{(x,k)} \right\}$$

As one can see $\overline{\mathbf{1}}_{(x,k)} = m_k(k, \bar{x}_k)$. What is important to see about these equivalence classes is that applying the substitution

$$\sigma_x = \left\{ y_0 \leftarrow x_0, \cdots, y_{i-1} \leftarrow x_{i-1}, y_i \leftarrow m_k(k, \bar{x}_k), y_{i+1} \leftarrow x_{i+1}, \cdots, y_k \leftarrow x_k \right\}$$

to a member of the class $\overline{\mathbf{n}}_{(y,k)}$ gives you a member of the class $\overline{\mathbf{n+1}}_{(x,k)}$. This is exactly the substitution used in the local unification of Lem. 7. We will use the equivalence classes in the resolution refutation to replace the terms in end-sequent formulae with the equivalence classes. When two formulae have different substitutions applied to them, but the resulting terms are in the same equivalence class, we contract the formulae. Also important to note is how to interpret a formula with equivalence classes replacing terms:

$$\vdash \varphi(\overline{\mathbf{n}}_{(x,k)}) \equiv \vdash \bigvee_{t \in \overline{\mathbf{n}}_{(x,k)}} \varphi(t)$$

$$\varphi(\overline{\mathbf{n}}_{(x,k)}) \vdash \equiv \bigwedge_{t \in \overline{\mathbf{n}}_{(x,k)}} \varphi(t) \vdash$$

The whole idea of the equivalence classes is to group together terms that state roughly the same thing.

The sequent proven derivable in Lem. 5 after adding the end-sequent formulae and equivalence classes is as follows:

$$\bigwedge_{i=0}^{k} f(x_{o(i)}) \sim o(i), \bigvee_{i=0}^{n} f(\overline{\mathbf{1}}_{(x,n)}) \sim i \vdash \bigvee_{i=0}^{k} (\overline{\mathbf{0}}_{(x,n)}^{o(i)} < \overline{\mathbf{1}}_{(x,n)} \wedge f(\overline{\mathbf{0}}_{(x,n)}^{o(i)}) \sim f(\overline{\mathbf{1}}_{(x,n)})), \bigvee_{i=k+1}^{n} f(m_n(n, \bar{x}_n)) \sim o(i)$$

The superscript on the equivalence classes $\overline{\mathbf{0}}_{(x,n)}$ denotes the index of the variable. Using the above sequent instead of the one derived in Lem. 5 in the proof of Lem. 7 and Thm. 1 we derive the following end sequent:

$$\bigwedge_{w=0}^{n+1} \bigvee_{i=0}^{n} f(\overline{\mathbf{w}}_{(x,n)}) \sim i \vdash \bigvee_{i=0}^{n} \bigvee_{w=i+1}^{n+1} (\bar{\mathbf{i}}_{(x,n)} < \overline{\mathbf{w}}_{(x,n)} \wedge f(\bar{\mathbf{i}}_{(x,n)}) \sim f(\overline{\mathbf{w}}_{(x,n)}))$$

Showing that the resolution refutation actually results in this end sequent requires a lot of work and would not fit in the bounds of this paper. However, intuitively we are just gathering the terms of Lem. 7 after substitution into sets based on a similarity condition (iterated max nesting). The main problem with the method outlined above is that it relies too much on the structure of this particular proof. However, part of the reason this proof was chosen in the first place to analyse the schematic CERES method is the redundancy and how it is at the limits of what the method can handle. We will discuss our plans dealing with application of this work in the next section.

What we have shown so far is how one can use schematic CERES to analyse a proof of a mathematical statement formalized in the $\mathbf{LKS}_{\varepsilon}$ calculus. We have also shown that the resolution calculus of [8] is too weak to express refutations of clause sets derived from fairly simple schematic proofs. The proof used in this paper, for example, only used two proofs links which have a simple call structure. Although we were able to construct a refutation in a mathematical meta-language, we have not yet been able to generalize the language of the schematic resolution refutation calculus. To circumvent this issue we extract a Herbrand sequent using the patterns found in the local unifiers of the proof of refutability. To get around the immense amount of repetition we construct equivalence classes which capture an important property of the term language, i.e. term depth. Using these equivalence classes we are able to extract the finite pigeon-hole principle.

5 Open Problem and Future Work

Many holes are left open in this work, most importantly the construction of a generalized language for the schematic resolution refutation calculus. Not only would the development of a language allow for ACNF construction, it could also help sharpen the method we introduce here for Herbrand sequent extraction. Also, we hope the construction of such a language would lead to a generalization of the Herbrand sequent extraction method as well, being that the method introduced here is very proof specific. Though, we expect that analysing the work showcased above will help lead to a generalization of the language, this was one of the motivations. For example, prior to the proof analysis of the tape proof, it was not clear that one would need all possible permutations of the values in the numeric sort. On a positive and unexpected note, only one free parameter exists in the resolution refutation and in the original language of the schematic resolution refutation calculus. It was considered at one point that multiple free parameters might occur in the resolution refutation of the proof.

Though constructing an ACNF has been the goal of proof analysis using CERES, in the schematic case having an ACNF does not provide as much information as expected, more correctly, it provides an overwhelming amount which is impossible to parse. Thus, we envision a move from constructing an ACNF at the end of schematic proof analysis to extraction of the Herbrand sequent. The main goal of future work will be to develop a method of Herbrand sequent extraction for a reasonable class of schematic proofs formalizable in $\mathbf{LKS}_\varepsilon$.

Acknowledgements: We would like to give special thanks to Sanja Ivkov [3] for Editorial help.

References

[1] M. Aigner & G. Ziegler (1999): *Proofs from THE BOOK*. Springer.

[2] Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter & Hendrik Spohr (2006): *Proof Transformation by CERES*. In JonathanM. Borwein & WilliamM. Farmer, editors: *Mathematical Knowledge Management*, *Lecture Notes in Computer Science* 4108, Springer Berlin Heidelberg, pp. 82–93.

[3] Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter & Hendrik Spohr (2008): *CERES: An analysis of Fürstenberg's proof of the infinity of primes*. *Theoretical Computer Science* 403, pp. 160–175.

[4] Matthias Baaz & Alexander Leitsch (2000): *Cut-elimination and redundancy-elimination by resolution*. *Journal of Symbolic Computation* 29, pp. 149–176.

[5] Matthias Baaz & Alexander Leitsch (2006): *Towards a clausal analysis of cut-elimination*. *Journal of Symbolic Computation* 41(3-4), pp. 381–410.

[6] David Baelde & Dale Miller (2007): *Least and greatest fixed points in linear logic*. In: *LPAR 2007*, *LNCS* 4790, pp. 92–106.

[7] James Brotherston (2005): *Cyclic Proofs for First-Order Logic with Inductive Definitions*. In B. Beckert, editor: *Automated Reasoning with Analytic Tableaux and Related Methods*, *Lecture Notes in Computer Science* 3702, pp. 78–92.

[8] Cvetan Dunchev, Alexander Leitsch, Mikheil Rukhaia & Daniel Weller (2012): *CERES for First-Order Schemata*. Http://arxiv.org/abs/1303.4257.

[9] Gerhard Gentzen (1935): *Untersuchungen über das logische Schließen I*. *Mathematische Zeitschrift* 39(1), pp. 176–210.

[10] Stefan Hetzl, Alexander Leitsch, Daniel Weller & Bruno Woltzenlogel Paleo (2008): *Herbrand sequent extraction*. In: *IN INTELLIGENT COMPUTER MATHEMATICS*, Springer, pp. 462–477.

[11] Raymond McDowell & Dale Miller (2000): *Cut-elimination for a logic with definitions and induction*. *Theoretical Computer Science* 232(1–2), pp. 91–119.

[12] Diana Ratiu & Trifon Trifonov (2012): *Exploring the Computational Content of the Infinite Pigeonhole Principle*. *J. Log. Comput.* 22(2), pp. 329–350. Available at `http://dx.doi.org/10.1093/logcom/exq011`.

[13] Christoph Sprenger & Mads Dam (2003): *On the Structure of Inductive Reasoning: Circular and Tree-Shaped Proofs in the μ-calculus*. In: *FOSSACS 2003*, *LNCS* 2620, pp. 425–440.

[14] William W. Tait (1968): *Normal derivability in classical logic*. In: *The Syntax and Semantics of Infinitary Languages*, *Lecture Notes in Mathematics* 72, Springer Berlin, pp. 204–236.

[15] Gaisi Takeuti (1975): *Proof Theory*. *Studies in logic and the foundations of mathematics* 81, American Elsevier Pub.

[3] http://about.me/sanja.ivkov