Resolution Refinements for Cut-Elimination based on Reductive Methods ⋆

Stefan Hetzl[2], Alexander Leitsch[1], Tomer Libal[1], Daniel Weller[1], and Bruno Woltzenlogel Paleo[1]

[1] {leitsch, shaolin, weller, bruno}@logic.at
Institute of Computer Languages (E185),
Vienna University of Technology,
Favoritenstraße 9, 1040 Vienna, Austria

[2] hetzl@lix.polytechnique.fr
INRIA Saclay –Île-de-France
École Polytechnique – LIX
91128 Palaiseau, France

Abstract. Traditional reductive cut-elimination and *CERES* seem to be methods of entirely different nature and hence hard to compare. This short paper describes ongoing research that aims at comparing and possibly combining them in ways that retain that best features of each method.

1 Introduction

Cut-elimination theorems and algorithms that actually perform the elimination of cuts from proofs are among the most prominent results and techniques of proof theory and of logic in general. Originally devised as a way to prove consistency [8], cut-elimination also plays a major role in: automated theorem proving, where the sub-formula property corollary allows a bottom-up construction of proofs; analysis of mathematical proofs, where the elimination of cuts corresponds to the elimination of undesired mathematical lemmas [2]; extraction of interpolants via Maehara's lemma, which requires cut-free proofs [11]; semantics and identity of proofs, where confluence of cut-elimination is important [10].

Therefore, it is important to compare different cut-elimination algorithms and devise new and hopefully better ones, as such improvements can potentially have implications for several areas of proof theory.

In this paper, in particular, we compare reductive cut-elimination methods and cut-elimination by resolution (*CERES*) (defined in Sections 2 and 3) and we propose a way to combine them via resolution refinements (Section 4). These refinements restrict the atomic cut normal forms (ACNFs, which are not cut-free, but whose cuts are at most atomic) obtainable by *CERES* essentially to those that are obtainable by reductive methods. The long range aim is to implement and use various refinements in the analysis of formalized mathematical proofs.

⋆ Supported by the Austrian Science Fund (project P19875)

2 Reductive Cut-Elimination Methods

The standard method of cut-elimination is that of Gentzen defined in his famous "Hauptsatz" [8]. The method is essentially a nondeterministic algorithm extracted from his (constructive) proof. Its characteristic feature is a rewriting system that rewrites proofs by shifting cut inferences upwards (rank reduction) and by reducing the complexity of cut-formulas when these are the main formulas of the inferences immediately above the cut (grade reduction). The result is a proof containing cuts that occur on top of the proof and whose cut-formulas are at most atomic. These atomic cuts can be simply eliminated, because their conclusion sequents are equal to their premise sequents.

3 *CERES*

The resolution-based method *CERES* for cut-elimination in classical logic has been defined in [6] and further developed in [3] and [9].

The method inductively defines a set of pairs (with a *clause* in the first component and a *projection* (to this clause) in the second component) C_ν for every node ν in a **skolemized**[3] proof φ:

- If ν is an occurrence of an axiom sequent $S(\nu)$, S' is the subsequent of $S(\nu)$ containing only the ancestors of cut-formula occurrences and S is the whole axiom, then $C_\nu = \{\langle S', S\rangle\}$.
- Let ν' be the predecessor of ν in a unary inference ρ.
  Let $C_{\nu'} = \{\langle c_1, \psi_1\rangle, \dots, \langle c_n, \psi_n\rangle\}$.
  (a) The auxiliary formulas of ν' are ancestors of cut-formula occurrences. Then
  $$C_\nu = C_{\nu'}$$
  (b) The auxiliary formulas of ν' are not ancestors of cut-formula occurrences. Then
  $$C_\nu = \{\langle c_1, \rho(\psi_1)\rangle, \dots, \langle c_n, \rho(\psi_n)\rangle\}$$
  where $\rho(\psi)$ denotes the derivation that is obtained from ψ by applying ρ to its end-sequent.
- Let ν_1, ν_2 be the predecessors of ν in a binary inference ρ.
  (a) The auxiliary formulas of ν_1, ν_2 are ancestors of cut-formula occurrences. Then
  $$C_\nu = C_{\nu_1} \cup C_{\nu_2}.$$
  (b) The auxiliary formulas of ν_1, ν_2 are not ancestors of cut-formula occurrences. Then
  $$C_\nu = C_{\nu_1} \times C_{\nu_2}.$$

[3] A skolemized proof is a proof with an end-sequent in skolem normal form. For any proof φ there is a proof φ' such that φ' is skolemized and the end-sequent of φ' is a structural skolem normal form of the end-sequent of φ [5].

where

$$\mathcal{C} \times \mathcal{D} = \{\langle c \circ d, \rho(\psi, \chi)\rangle \mid \langle c, \psi\rangle \in C, \langle d, \chi\rangle \in D\}$$

where $c \circ d$ is the merge of clauses and $\rho(\psi, \chi)$ denotes the derivation that is obtained from the derivations ψ and χ by applying the binary inference ρ.

The *characteristic clause set* CL(φ) of φ is defined as $\mathcal{C}_{\nu_0}$, where ν_0 is the root.

CL(φ) is always unsatisfiable [6]. Therefore, there is a resolution refutation of CL(φ), which can be grounded and then used as a skeleton where each leaf clause receives its corresponding instantiated projection on top. Finally, the resolution and factoring inferences can be replaced by cuts and contractions, respectively, yielding a proof of the end-sequent of φ in ACNF (possibly with the addition of contractions in the bottom of the proof).

The main advantage of *CERES* over reductive cut-elimination methods is that it is implicitly capable of detecting redundancies in the input proof φ and eliminating them. Therefore, *CERES* can, in the best cases, produce ACNFs that are non-elementarily smaller than ACNFs produced by reductive methods [6]. More precisely, there are proofs φ such that, for all ACNFs $\varphi_{\triangleright_{GT}}$ obtained via Gentzen's or Tait's reductive cut-elimination methods, there exists an ACNF $\varphi_{\mathbf{CERes}}$ obtained via *CERES* such that:

$$\frac{|\varphi_{\triangleright_{GT}}|}{|\varphi_{\mathbf{CERes}}|} = O(\overbrace{2^{2^{2^{\cdots}}}}^{|\varphi|})$$

where $|\psi|$ denotes the size of the proof ψ.

For this reason, *CERES* is computationally superior than reductive cut-elimination methods, especially considering that the converse (i.e. proofs whose ACNFs via reductive cut-elimination would be non-elementary smaller than ACNFs via *CERES*) is not possible [6]. However, the price paid by *CERES* is its increased non-confluence (i.e. *CERES* can usually produce more ACNFs than reductive cut-elimination methods) and a correspondingly large search space for refutations. In some cases, such as for Fuerstenberg's proof of the infinitude of primes, current theorem provers like Otter and Prover9 were unable to refute the characteristic clause set [4].

4 Resolution Refinements for Cut-Elimination

In order to tackle the problem of the large search space for refutations, the chosen approach was the development of resolution refinements that reduce the number of *CERES* ACNFs that are not reductive ACNFs[4].

[4] Strictly speaking, ACNFs produced by CERes are structurally very different from ACNFs produced by reductive cut-elimination methods. In the former the atomic cuts occur in the bottom, while in the latter they occur in the top of the ACNF. However, the ACNFs can be compared with respect to their *canonic refutation* [7]. In this paper,

The following examples show some kinds of proofs whose characteristic clause sets admit refutations that lead to ACNFs that are not obtainable with reductive cut-elimination methods. Each example motivates the development of a different refinement.

4.1 Blocking the Resolution of Literals from Different Cuts

Consider the proof φ below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P\alpha \vdash P\alpha}{P\alpha, \neg P\alpha \vdash P\alpha}\,w_l}{P\alpha \vdash \neg P\alpha \rightarrow P\alpha}\,\rightarrow_r}{\forall x P x \vdash \neg P\alpha \rightarrow P\alpha}\,\forall_l}{\forall x P x \vdash \forall x(\neg P x \rightarrow P x)}\,\forall_r \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Ps \vdash Ps}{\vdash \neg Ps, Ps}\,\neg_r \quad Ps \vdash Ps}{\neg Ps \rightarrow Ps \vdash Ps, Ps}\,\rightarrow_l}{\neg Ps \rightarrow Ps \vdash Ps}\,c_r}{\forall x(\neg P x \rightarrow P x) \vdash Ps}\,\forall_l \qquad \dfrac{Ps \vdash Ps}{Ps \vdash \exists y P y}\,\exists_r}{\forall x(\neg P x \rightarrow P x) \vdash \exists y P y}\,cut}{\forall x P x \vdash \exists y P y}\,cut$$

Its characteristic clause set is:

$$\mathrm{CL}(\varphi) \equiv \{\underbrace{\vdash P\alpha}_{c_1};\underbrace{Ps \vdash Ps}_{c_2};\underbrace{Ps \vdash Ps}_{c_3};\underbrace{Ps \vdash}_{c_4}\}$$

This characteristic clause admits the following resolution refutation δ:

$$\dfrac{c_1 \qquad c_4}{\vdash}\,R$$

δ is used as a skeleton for an ACNF that will have an atomic cut where the cut formula occurrences are Ps from c_4 and the instance Ps from the occurrence $P\alpha$ from c_1. But, if reductive cut-elimination methods had been used, this could not happen, because reductive cut-elimination methods are local. As the cuts are shifted upwards, grade reduction always keeps cut-formula occurrences of a cut paired (via the new cuts) with cut-formula occurrences of the same original cut. But the literals of c_1 come from ancestors of the lowermost cut, while the literals of c_4 come from ancestors of the uppermost cut. δ is effectively pairing cut formula occurrences from different cuts.

In order to prevent this class of refutations exemplified by δ, the ancestors of cut formula occurrences can be annotated with labels in such a way that two occurrences have the same label iff they are *cut-linked*.

Definition 1 (Cut-Linkage). *Two (sub)formula occurrences ν_1 and ν_2 in a proof φ are* cut-linked *if and only if there is a cut ρ such that ν_1 is an ancestor of ν_i and ν_2 is an ancestor of ν_j where ν_i and ν_j are auxiliary occurrences of ρ.*

The labels for cut-linkage in φ are shown below:

two ACNFs obtained from the same input proof are considered equal if they have the same canonic refutation.

$$\begin{prooftree}\AxiomC{φ_1}\AxiomC{φ_2}\RightLabel{cut}\BinaryInfC{$\forall x Px \vdash \exists y Py$}\end{prooftree}$$

where φ_1 is:

$$\begin{prooftree}\AxiomC{$P\alpha \vdash [P\alpha]_1$}\RightLabel{$w_l$}\UnaryInfC{$P\alpha, \neg[P\alpha]_1 \vdash [P\alpha]_1$}\RightLabel{$\rightarrow_r$}\UnaryInfC{$P\alpha \vdash [\neg P\alpha \rightarrow P\alpha]_1$}\RightLabel{$\forall_l$}\UnaryInfC{$\forall x Px \vdash [\neg P\alpha \rightarrow P\alpha]_1$}\RightLabel{$\forall_r$}\UnaryInfC{$\forall x Px \vdash \forall x[(\neg Px \rightarrow Px)]_1$}\end{prooftree}$$

and φ_2 is:

$$\begin{prooftree}\AxiomC{$[Ps]_1 \vdash [Ps]_2$}\RightLabel{$\neg_r$}\UnaryInfC{$\vdash \neg[Ps]_1, [Ps]_2$}\AxiomC{$[Ps]_1 \vdash [Ps]_2$}\RightLabel{$\rightarrow_l$}\BinaryInfC{$[\neg Ps \rightarrow Ps]_1 \vdash [Ps]_2, [Ps]_2$}\RightLabel{$c_r$}\UnaryInfC{$[\neg Ps \rightarrow Ps]_1 \vdash [Ps]_2$}\RightLabel{$\forall_l$}\UnaryInfC{$\forall x[(\neg Px \rightarrow Px)]_1 \vdash [Ps]_2$}\AxiomC{$[Ps]_2 \vdash Ps$}\RightLabel{$\exists_r$}\UnaryInfC{$[Ps]_2 \vdash \exists y Py$}\RightLabel{$cut$}\BinaryInfC{$\forall x[(\neg Px \rightarrow Px)]_1 \vdash \exists y Py$}\end{prooftree}$$

The new characteristic clause set is essentially the same as before, but now the literals have the labels that indicate the cuts from which they originate:

$$\mathrm{CL}(\varphi) \equiv \{\underbrace{\vdash [P\alpha]_1}_{c_1}; \underbrace{[Ps]_1 \vdash [Ps]_2}_{c_2}; \underbrace{[Ps]_1 \vdash [Ps]_2}_{c_3}; \underbrace{[Ps]_2 \vdash}_{c_4}\}$$

We define *R_{cl}-resolution* as resolution restricted in such a way that two literals can only be resolved if they have the same labels (i.e. if they originated from the same cut). It is clear that δ is not an R_{cl}-refutation.

4.2 Blocking the Resolution of Literals from the Same Branch of a Cut

The previously described refinement of R_{cl}-refutation still can produce refutations whose corresponding ACNFs would not be obtainable by reductive cut-elimination methods. This can occur, for example, when the proof contains only one cut but the cut-formula is valid, as shown in the proof φ below:

$$\begin{prooftree}\AxiomC{$P\alpha \vdash [P\alpha]_1$}\RightLabel{$w_r$}\UnaryInfC{$P\alpha \vdash [\neg P\alpha]_1, [P\alpha]_1$}\RightLabel{$\vee_r$}\UnaryInfC{$P\alpha \vdash [\neg P\alpha \vee P\alpha]_1$}\RightLabel{$\forall_l$}\UnaryInfC{$\forall x Px \vdash [\neg P\alpha \vee P\alpha]_1$}\RightLabel{$\forall_r$}\UnaryInfC{$\forall x Px \vdash [\forall x(\neg Px \vee Px)]_1$}\AxiomC{$Pt \vdash [Pt]_1$}\RightLabel{$\neg_l$}\UnaryInfC{$[\neg Pt]_1, Pt \vdash$}\RightLabel{$\neg_r$}\UnaryInfC{$[\neg Pt]_1 \vdash \neg Pt$}\AxiomC{$[Pt]_1 \vdash Pt$}\RightLabel{$\vee_l$}\BinaryInfC{$[\neg Pt \vee Pt]_1 \vdash Pt, \neg Pt$}\RightLabel{$\vee_r$}\UnaryInfC{$[\neg Pt \vee Pt]_1 \vdash Pt \vee \neg Pt$}\RightLabel{$\forall_l$}\UnaryInfC{$[\forall x(\neg Px \vee Px)]_1 \vdash Pt \vee \neg Pt$}\RightLabel{$cut$}\BinaryInfC{$\forall x Px \vdash Pt \vee \neg Pt$}\end{prooftree}$$

Its characteristic clause set is:

$$\mathrm{CL}(\varphi) \equiv \{\underbrace{\vdash [P\alpha]_1}_{c_1}; \underbrace{\vdash [Pt]_1}_{c_2}; \underbrace{[Pt]_1 \vdash}_{c_3}\}$$

Note that all clauses of the set have the same label, because φ has only one cut. This means that, for this case, any unrestricted R-refutation would also be an R_{cl}-refutation. Hence R_{cl}-resolution does not really help in this case.

Let δ be the R_{cl}-refutation below:

$$\dfrac{c_2 \qquad c_3}{\vdash}\, R_{cl}$$

Both c_2 and c_3 contain literals originating from the right branch of the cut. By executing reductive cut-elimination on φ, on the other hand, the final atomic cut will necessarily have an instance of $P\alpha$ and the Pt from the second branch of the $\vee_l$ rule as its cut-formula occurrences. In general, in reductive cut-elimination methods the atomic cuts of the resulting ACNF must pair occurrences originating from different branches of the original cuts. This is so, because in the grade reduction rewrite rules, it is never the case that the new cuts pair occurrences that are subformulas of the same cut-formula occurrences.

To prevent refutation as δ above, *side-labels* l and r can be added to the ancestors of cut formula occurrences, indicating whether they are ancestor from the left or from the right cut formula occurrence. This is shown below:

$$\begin{prooftree}
\AxiomC{$P\alpha \vdash [P\alpha]^l_1$}
\RightLabel{w_r}
\UnaryInfC{$P\alpha \vdash [\neg P\alpha]^l_1, [P\alpha]_1$}
\RightLabel{$\vee_r$}
\UnaryInfC{$P\alpha \vdash [(\neg P\alpha \vee P\alpha)]^l_1$}
\RightLabel{$\forall_l$}
\UnaryInfC{$\forall x Px \vdash [(\neg P\alpha \vee P\alpha)]^l_1$}
\RightLabel{$\forall_r$}
\UnaryInfC{$\forall x Px \vdash [\forall x(\neg Px \vee Px)]^l_1$}
\AxiomC{$Pt \vdash [Pt]^r_1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$[\neg Pt]^r_1, Pt \vdash$}
\RightLabel{$\neg_r$}
\UnaryInfC{$[\neg Pt]^r_1 \vdash \neg Pt$}
\AxiomC{$[Pt]^r_1 \vdash Pt$}
\RightLabel{$\vee_l$}
\BinaryInfC{$[(\neg Pt \vee Pt)]^r_1 \vdash Pt, \neg Pt$}
\RightLabel{$\vee_r$}
\UnaryInfC{$[(\neg Pt \vee Pt)]^r_1 \vdash Pt \vee \neg Pt$}
\RightLabel{$\forall_l$}
\UnaryInfC{$[\forall x(\neg Px \vee Px)]^r_1 \vdash Pt \vee \neg Pt$}
\RightLabel{cut}
\BinaryInfC{$\forall x Px \vdash Pt \vee \neg Pt$}
\end{prooftree}$$

$$\mathrm{CL}(\varphi) \equiv \{\underbrace{\vdash [P\alpha]^l_1}_{c_1}; \underbrace{\vdash [Pt]^r_1}_{c_2}; \underbrace{[Pt]^r_1 \vdash}_{c_3}\}$$

R_{cls}-resolution is defined as R_{cl}-resolution with the additional constraint that two literals can only be resolved if their side labels are different. From the side-labelled $\mathrm{CL}(\varphi)$ above, one can see that the R_{cl}-refutation δ is not an R_{cls}-refutation, because the literals of c_2 and c_3 have the same side-label r and hence cannot be resolved with each other.

4.3 Blocking the Resolution of Literals from Different Positions in Cuts

Still it is possible to have R_{cls}-refutations whose corresponding ACNFs are not obtainable via reductive cut-elimination methods. This fact can be exemplified by the proof φ below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P\alpha \vdash [P\alpha]^l_1}{\forall zPz \vdash [P\alpha]^l_1}\,\forall_l}{\forall zPz \vdash [\forall xPx]^l_1}\,\forall_r}{\forall zPz \vdash [\forall xPx]^l_1, [\forall yPy]^l_1}\,w_r}{\forall zPz \vdash [(\forall xPx \lor \forall yPy)]^l_1}\,\lor_r \qquad \dfrac{\dfrac{\dfrac{\dfrac{[Pt]^r_1 \vdash Pt}{[Pt]^r_1 \vdash \exists wPw}\,\exists_r}{[\forall xPx]^r_1 \vdash \exists wPw}\,\forall_l \quad \dfrac{\dfrac{[Ps]^r_1 \vdash Ps}{[Ps]^r_1 \vdash \exists wPw}\,\exists_r}{[\forall yPy]^r_1 \vdash \exists wPw}\,\forall_l}{[(\forall xPx \lor \forall yPy)]^r_1 \vdash \exists wPw, \exists wPw}\,\lor_l}{[(\forall xPx \lor \forall yPy)]^r_1 \vdash \exists wPw}\,c_r}{\forall zPz \vdash \exists wPw}\,cut$$

Its characteristic clause set is:

$$\mathrm{CL}(\varphi) \equiv \{\underbrace{\vdash [P\alpha]^l_1}_{c_1}; \underbrace{[Pt]^r_1 \vdash}_{c_2}; \underbrace{[Ps]^r_1 \vdash}_{c_3}\}$$

And it admits the following R_{cls}-refutation δ:

$$\dfrac{c_1 \qquad c_3}{\vdash}\,R_{cls}$$

No ACNF produced by reductive cut-elimination would have an atomic cut whose cut formula occurrences come from $P\alpha$ and Ps. Instead, $P\alpha$ would be resolved with Pt, because $P\alpha$ and Pt originate from the left disjunct of the cut-formula, while Ps originates from the right disjunct, and grade reduction mantains this structure when it creates new cuts of smaller formula complexity.

To forbid refutations like δ, a more strict labeling, called atomic cut linkage, of the cut-formula ancestors can be devised.

Definition 2 (Atomic Cut-Linkage). *Two atomic (sub)formula occurrences ν_1 and ν_2 in a proof φ are* atomically cut-linked *if and only if there is a cut* ρ *such that ν_1 is an ancestor of $\lfloor \nu_i \rfloor_\pi$ and ν_2 is an ancestor of $\lfloor \nu_j \rfloor_\pi$ where π is the position of an atomic sub-formula and ν_i and ν_j are auxiliary occurrences of* ρ.

In the proof below, atomic subformula occurrences of cut ancestors are given labels such that if two occurrences have the same label, then they are atomic cut-linked:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P\alpha \vdash [P\alpha]_1}{\forall zPz \vdash [P\alpha]_1}\,\forall_l}{\forall zPz \vdash \forall x[Px]_1}\,\forall_r}{\forall zPz \vdash \forall x[Px]_1, \forall y[Py]_2}\,w_r}{\forall zPz \vdash (\forall x[Px]_1 \lor \forall y[Py]_2)}\,\lor_r \qquad \dfrac{\dfrac{\dfrac{\dfrac{[Pt]_1 \vdash Pt}{[Pt]_1 \vdash \exists wPw}\,\exists_r}{\forall x[Px]_1 \vdash \exists wPw}\,\forall_l \quad \dfrac{\dfrac{[Ps]_2 \vdash Ps}{[Ps]_2 \vdash \exists wPw}\,\exists_r}{\forall y[Py]_2 \vdash \exists wPw}\,\forall_l}{(\forall x[Px]_1 \lor \forall y[Py]_2) \vdash \exists wPw, \exists wPw}\,\lor_l}{(\forall x[Px]_1 \lor \forall y[Py]_2) \vdash \exists wPw}\,c_r}{\forall zPz \vdash \exists wPw}\,cut$$

The characteristic clause set with the atomic cut-linkage labels is:

$$\mathrm{CL}(\varphi) \equiv \{\underbrace{\vdash [P\alpha]_1}_{c_1}; \underbrace{[Pt]_1 \vdash}_{c_2}; \underbrace{[Ps]_2 \vdash}_{c_3}\}$$

R_{acl}-resolution is defined as R-resolution with the restriction that two literals can only be resolved if they have the same atomic cut-linkage label. Clearly, as desired, δ is not an R_{acl}-refutation.

We can now give a uniform definition of the refined resolution rules

Definition 3 (Refined Resolution and Factoring Rules). *The* resolution rule R *shown below:*

$$. \frac{\Gamma_1 \vdash \Delta_1, A_1 \qquad A_2, \Gamma_2 \vdash \Delta_2}{(\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2) mgu(A_1, A_2)} R$$

where $\Gamma_1 \vdash \Delta_1, A_1$ and $A_2, \Gamma_2 \vdash \Delta_2$ are variable-disjoint clauses, is a:

– Cut-Linkage Refined Resolution Rule R_{cl} *iff A_1 and A_2 are cut-linked.*
– Cut-Linkage/Sides Refined Resolution Rule R_{cls} *iff A_1 and A_2 are cut-linked and from opposite sides (branches) of a cut.*
– Atomic Cut-Linkage Refined Resolution Rule R_{acl} *iff A_1 and A_2 are atomically cut-linked.*

Analogously, the restricted rules of factoring *should also be restricted so that, if $A_1, \ldots, A_n$ are the factorized atoms, then the factoring is a:*

– Cut-Linkage Factoring F_{cl} *iff $A_1, \ldots, A_n$ are pairwise cut-linked.*
– Cut-Linkage/Sides Factoring F_{cls} *iff $A_1, \ldots, A_n$ are pairwise cut-linked and from the same side (branch) of a cut.*
– Atomic Cut-Linkage Refined Resolution Rule F_{acl} *iff $A_1, \ldots, A_n$ are pairwise atomically cut-linked.*

5 Refined Refutability

The original proof of the refutability of the characteristic clause set shows that the characteristic clause set is unsatisfiable by constructing a refutation in sequent calculus **LK** [6]. Then it relies on the completeness of the unrestricted resolution calculus, which guarantees that refutations of unsatisfiable clause sets exist.

However, with the restrictions imposed by the refinements, one cannot rely on the completeness of resolution anymore. Indeed, for *arbitrary* clause sets with *arbitrary* labels, R_{cl}-resolution, R_{cls}-resolution and R_{acl}-resolution are clearly incomplete. Nevertheless, Theorem 1 shows that for *characteristic* clause sets extracted from proofs with the labeling done in the *specific* ways defined in Section 4, refined refutations always exist.

Theorem 1 (Refutability of the Characteristic Clause Set). *For any proof φ,* CL(φ) *is R-refutable, R_{cl}-refutable, R_{cls}-refutable and R_{acl}-refutable.*

Proof. The full detailed proof is under development in the unfinished PhD thesis of Bruno Woltzenlogel Paleo. What follows is a basic informal outline of the ideas of the proof.

Firstly, it can be noted that every R_{acl}-refutation is an R_{cls}-refutation, every R_{cls}-refutation is an R_{cl}-refutation, and every R_{cl}-refutation is an R-refutation. Hence, it suffices to show that $\mathrm{CL}(\varphi)$ is R_{acl}-refutable.

We prove the refutability by constructing an R_{acl}-refutation, as follows:

– Let φ' be an atomic cut normal form of φ obtained by reductive methods (i.e. by applying rank and grade reduction, but no elimination of atomic cuts).
– Lemma 1 (Subsumption of the Characteristic Clause Sets under cut reduction):
  Show that $\mathrm{CL}(\varphi')$ is subsumed by $\mathrm{CL}(\varphi)$ [7].
– Lemma 2 (Invariance of the Atomic Cut Linkage Labeling under cut-reduction): Show that, when a cut reduction is performed, the labels of the cut-formula occurrences of the new cuts are the same, so that in φ', the atomic cuts always resolve two atoms that have the same labels.
– Lemma 3 (Canonic Refutation):
  Show that $\mathrm{CL}(\varphi')$ admits a canonic refutation [7], which can be extracted from φ' roughly by taking the cut-relevant part of φ', which is composed of atomic cut inferences only, and transforming these atomic cuts into resolution inferences. Let δ' be this canonic refutation.
– By Lemma 1, δ' can be lifted to a R-refutation δ of $\mathrm{CL}(\varphi)$, because $\mathrm{CL}(\varphi)$ subsumes $\mathrm{CL}(\varphi')$.
– By Lemma 2, δ is an R_{acl}-refutation of $\mathrm{CL}(\varphi)$.

6 Conclusions and Future Work

The refinements defined in this paper correspond, in various degrees, to the simulation of reductive methods within *CERES*. This allows a tradeoff between confluence of cut-elimination and size of the ACNFs, and consequently also some control on the search space for refutations.

The structural differences between ACNFs produced by *CERES* and reductive methods indicate some possible directions for future work in this area. It is noticeable that an ACNF produced by *CERES* ends with a series of contraction inferences. The duplications of formula occurrences (which are eventually contracted in the end) occur because of three different reasons:

1. Duplications of subproofs are intrinsic to the proccess of (reductive) cut-elimination in classical logics (due to rank reduction over contraction inferences).
2. Parts of the input proof are duplicated to appear in many projections. This is necessary to allow projections to be plugged on top of the refutation of the characteristic clause set. It seems that the contractions that exist due to this source of duplications could be avoided either by a more careful

construction and combination of the projections with the refutation or by a postprocessing step in which the contractions would be shifted upwards until they meet the weakening inferences that introduce one of their auxiliary formula occurrences, in which case both the contraction and the weakening could be eliminated.

3. The construction of the characteristic clause set may be seen as a standard CNF-transformation of the *characteristic clause term* [7]. The standard CNF transformation (which distributes disjunction over conjunction, or in this case products over sums in the charateristic clause term) can cause an exponential increase in size. The contractions associated with this source of duplications are intrinsic to cut-elimination by resolution using standard CNF-transformation, but it might be fruitful to investigate the possibility of using structural CNF-transformations, for which these duplications do not occur [1]. However, this approach would imply a radical change in the concepts of characteristic clause set and projections, because structural CNF-transformation adds fresh predicate symbols to the signature of the input proof.

An improvement of *CERES* that eliminates the second and third sources of contractions mentioned above would not only improve the efficiency of the method but also make it suitable for substructural logics in which contraction rules are not available. A deeper understanding of the relation between *CERES* and reductive methods seems to be crucial to achieve this improvement.

References

1. M. Baaz, U. Egly, and A. Leitsch. Normal form transformations. In A. Voronkov A. Robinson, editor, *Handbook of Automated Reasoning*, pages 275–333. Elsevier, 2001.
2. Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter, and Hendrik Spohr. Cut-Elimination: Experiments with CERES. In Franz Baader and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR) 2004*, volume 3452 of *Lecture Notes in Computer Science*, pages 481–495. Springer, 2005.
3. Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter, and Hendrik Spohr. Proof Transformation by CERES. In Jonathan M. Borwein and William M. Farmer, editors, *Mathematical Knowledge Management (MKM) 2006*, volume 4108 of *Lecture Notes in Artificial Intelligence*, pages 82–93. Springer, 2006.
4. Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter, and Hendrik Spohr. Ceres: An analysis of fürstenberg's proof of the infinity of primes. *Theor. Comput. Sci.*, 403(2-3):160–175, 2008.
5. Matthias Baaz and Alexander Leitsch. Cut normal forms and proof complexity. *Annals of Pure and Applied Logic*, 97:127–177, 1999.
6. Matthias Baaz and Alexander Leitsch. Cut-elimination and Redundancy-elimination by Resolution. *Journal of Symbolic Computation*, 29(2):149–176, 2000.
7. Matthias Baaz and Alexander Leitsch. Towards a clausal analysis of cut-elimination. *Journal of Symbolic Computation*, 41:381–410, 2006.
8. G. Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210,405–431, 1934–1935.

9. Stefan Hetzl. *Characteristic Clause Sets and Proof Transformations*. PhD thesis, Vienna University of Technology, 2007.
10. Lutz Straßburger. What is a logic, and what is a proof? In Jean-Yves Beziau, editor, *Logica Universalis*, pages 135–145. Birkhäuser, 2005. Updated version at http://www.lix.polytechnique.fr/~lutz/papers/WhatLogicProof.pdf.
11. G. Takeuti. *Proof Theory*. North-Holland, Amsterdam, 2 edition, 1987.