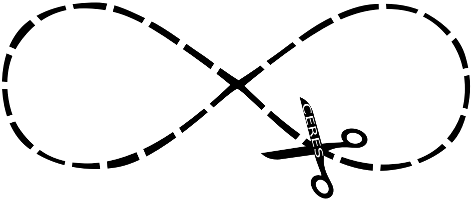
Cut-Elimination in Schematic Proofs and Herbrand Sequent Extraction



David M. Cerna

November 10, 2014

Goals

- Explain the basics of schematic cut-elimination using the CERES method (Dunchev et al. 2012), as well as the motivation behind the development of the method.
- Show the limitations of the method by providing an example formal proof whose clause set does not have a schematic refutation expressible in the resolution calculus of (Dunchev et al. 2012).
- Without a schematic resolution refutation, we cannot algorithmically construct an Atomic Cut Normal Form (ACNF). We show that even without an ACNF, we are able to extract a Herbrand sequent from a proof that the clause set is schematically refutable.

Why have a Schematic Cut-elimination Method?

- As it was shown in (Dunchev et al. 2012), The LK calculus extended with an induction rule, does not admit reductive cut-elimination. There exists cut formulae which cannot be passed over the induction rule, essentially cut formulae containing the eigenvariable introduced by the induction rule.
- Instead of adding an induction rule to the LK calculus, a new calculus was introduced in (Dunchev et al. 2012), the $\mathbf{LKS}_{\mathcal{E}}$, defining proofs primitive recursively with a free parameter over proofs.

Example $\mathbf{LKS}_{\mathcal{E}}$ Derivation

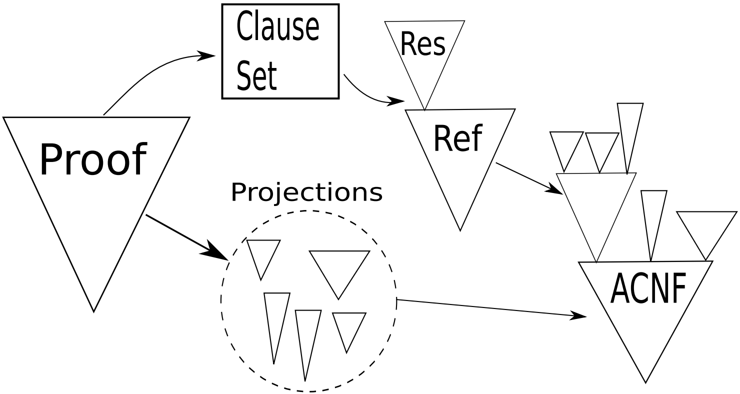
Stepcase ω

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

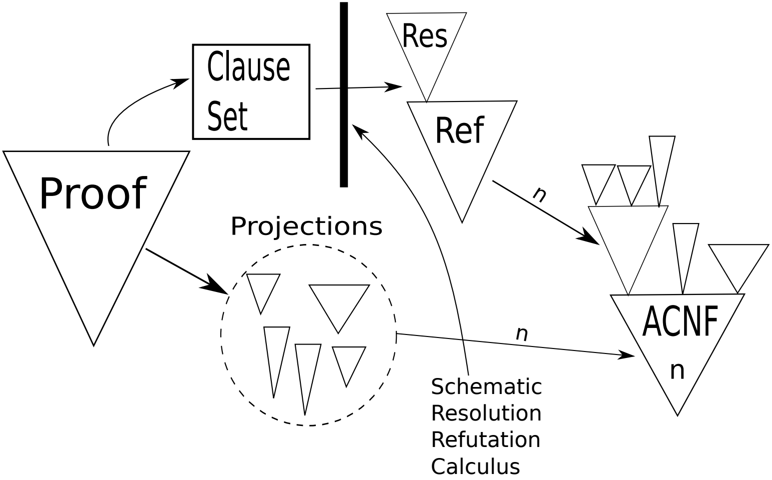
Basecase ω

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash \alpha \leq \alpha \qquad f(\alpha) \sim 0 \vdash f(\alpha) \sim 0}{f(\alpha) \sim 0 \vdash \alpha \leq \alpha \wedge f(\alpha) \sim 0}\ \wedge : r}{f(\alpha) \sim 0 \vdash \exists y (\alpha \leq y \wedge f(y) \sim 0)}\ \exists : r}{\forall x f(x) \sim 0 \vdash \exists y (\alpha \leq y \wedge f(y) \sim 0)}\ \forall : l}{\forall x f(x) \sim 0 \vdash \forall x \exists y (x \leq y \wedge f(y) \sim 0)}\ \forall : r \qquad \dfrac{\color{red}{\mu(0)}}{\forall x \exists y (x \leq y \wedge f(y) = 0) \vdash \exists p \exists q (p < q \wedge f(p) \sim f(q))}}{\forall x f(x) \sim 0 \vdash \exists p \exists q (p < q \wedge f(p) \sim f(q))}\ cut$$

CERES Method in a Nutshell



CERES$_s$ Method in a Nutshell



Distinction Between First-order and Schematic CERES

- On the two previous slides, the methods are shown nearly identical in the abstract, except for two minor details:

1) We need to instantiate the free parameter to some value of n in order to construct the ACNF from the projections and resolution refutation.
2) Only resolution refutations which fit the constraints of the schematic resolution refutation calculus (SRRC) can be considered (Dunchev et al. 2012).

- The decision problem, does a given clause set (which is known to refutable) have a schematic resolution refutation obeying the constraints of the SRRC, is undecidable.

Constraints of the SRRC

- Currently we do not have precise constraints on the expressive power of the SRRC.
- However, every schematic formal proof which has a refutation expressible in the SRRC, Including formal proofs with non-elementary length, have had term languages with at most monadic function symbols.
- The proof we formalize in this work contains a binary function symbol. A problem with the inclusion of a binary function symbol is unification can now add a finite number of extra variables by nesting the function symbol.
- These extra variables are introduced because they will be unified at some point with terms found on other branches of the proof.

Without a Refutation Which is Expressible by the SRRC

- When one does not have such a refutation of the clause set, the ACNF is no longer algorithmically constructable.
- To fix the problem, one could build a stronger language for the schematic resolution refutations; we are currently working on this problem.
- However, if one has a proof that the clause set is refutable for every instance, one could try to extract a schematic Herbrand sequent instead. This is what we will provide in this work.

Herbrand Sequent

- A Herbrand sequent is a generalization of a Herbrand disjunction for usage within a sequent calculus.
- Consider it as containing the creative content of the formal proof, i.e. the needed term instantiation for weak quantifiers. See (Hetzl et al. 2008).
- It is propositionally valid modulo the axioms of the underlying theory. In our case, we have additional axioms for equality and the maximum function.
- A schematic Herbrand sequent can include predicates and terms indexed by a single free parameter.

Non-injectivity Assertion and Principal Cut

Statement (Non-injectivity Assertion (NiA))

Let $f : \mathbb{N} \to [0, \cdots, n]$, where $n \in \mathbb{N}$, be total, then there exists $i, j \in \mathbb{N}$ such that $i < j$ and $f(i) \sim f(j)$.

Lemma (Infinity Lemma)

Given a total function $f : \mathbb{N} \to \mathbb{N}_{n+1}$ then either for all $x \in \mathbb{N}$ there exists a $y \in \mathbb{N}$ such that $x \leq y$ and $f(y) \in \mathbb{N}_n$, or for all $x \in \mathbb{N}$ there exists a $y \in \mathbb{N}$ such that $x \leq y$ and $f(y) \sim n + 1$.

- We do not provide the full formal proof here being that it is not the main focus of this work.

Axioms

- We use the following axioms to prove the statement:

$$\bigwedge_{i=0}^{n+1} \forall x, y((f(x) \sim i \wedge f(y) \sim i) \rightarrow f(x) \sim f(y)) \tag{1a}$$

$$\forall x, y(x < y \rightarrow s(x) \leq y) \tag{1b}$$

$$\forall x(x \leq x) \tag{1c}$$

$$\forall x, y, z(max(x, y) \leq z \rightarrow x \leq z) \tag{1d}$$

$$\forall x, y, z(max(x, y) \leq z \rightarrow y \leq z) \tag{1e}$$

Schematic Clause set

The following sequents represent the schematic clause set $C(n)$ for $n \in \mathbb{N}$:

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

Schematic Max Term

- The clauses (C1), (C2), and (C3) can generate arbitrarily long chains of max function nesting. These chains are essential to the refutation and require us to create schematic arity function symbols.
- The following definition will be used to refute the clause set:

Definition

Let us define the term $m_n(k, \bar{x}_n)$ where $k, n \in \mathbb{N}$ and $\bar{x}_n = x_0, \cdots, x_n$, as follows:

$$When\ n < k+1, \quad m_n(k+1, \bar{x}_n) \Rightarrow m_n(k, \bar{x}_n) \tag{2a}$$

$$When\ k+1 \leq n, \quad \begin{array}{c} m_n(k+1, \bar{x}_n) \Rightarrow \\ max(m_n(k, \bar{x}_n), s(x_{k+1})) \end{array} \tag{2b}$$

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

Overview of the Refutation

- The schematic refutation of C(n) require a complex induction of which we would only like to highlight the effect of using a binary function symbol.
- We need to define a well ordering specific to the resolution derivations of the clause in order to refute the clause set for every value of n.
- We need to use the special max term from the previous slide and match variable positions to symbols of the schematic sort. This is where the inter-dependency between branches arises.

Basecases for Well Ordering

Lemma (Greatest Lower Bounds)

Given $0 \leq n$, $-1 \leq k \leq n$ *and for all bijective functions* $b : \mathbb{N}_n \to \mathbb{N}_n$. *the formula*

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

is derivable from $C(n)$.

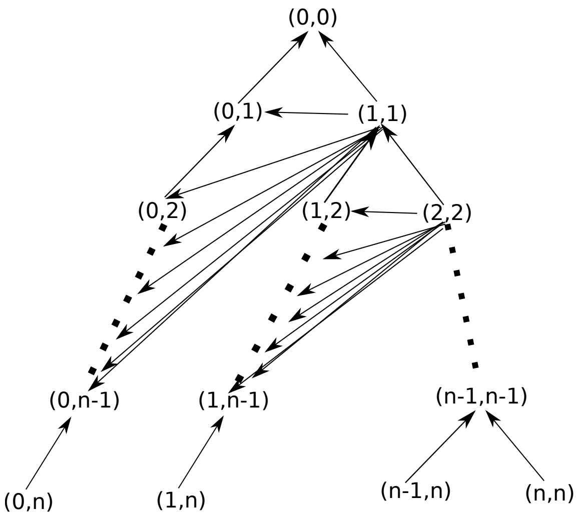
Lemma (General Members)

Given $0 \leq k \leq j \leq n$, *for all bijective functions* $b : \mathbb{N}_n \to \mathbb{N}_n$ *the clause*

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

is derivable from $C(n)$.

Well Ordering $\lessdot_n$



Notes on Induction (part 2)

- This inter-dependency of terms between branches requires us to take count of which subset of variable positions is in use at each point of the proof and in which order they will be used.
- Starting from the position (n, n) in the ordering, we can see that this is analogous to the subset sum, i.e.

$$n! \cdot \sum_{i=0}^{n} \frac{1}{i!} \approx e \cdot n!$$

- Even though the SRRC was designed to handle refutations with non-elementary length, the refutations in these cases did not have inter-dependency of branches, i.e. if one branch does not contain the n^{th} symbol and the other does, than we know that further up the branch the n^{th} symbol will be added.

Herbrand Sequent Extraction

- The idea is to add the projections without the weak quantifier instantiations to the proper clauses of the clause set. This is very specific to the proof we are working with here.
- Then we put the terms which are constructable from the unifications used in the proof of refutability into equivalence classes.
- We build the equivalence classes in such a way that they are ordered and unification takes one equivalence class and makes it into the next one in the ordering.

Clause set with End Sequent Literals

$(C'1)$	$\alpha \leq \alpha$
$(C'2)$	$max(\alpha, \beta) \leq \gamma \vdash \alpha \leq \gamma$
$(C'3)$	$max(\alpha, \beta) \leq \gamma \vdash \beta \leq \gamma$
$(C'4_0)$	$f(\alpha) \sim 0, f(\beta) \sim 0, s(\beta) \leq \alpha \vdash \beta < \alpha \wedge f(\alpha) \sim f(\beta)$
$\vdots$	$\vdots$
$(C'4_n)$	$f(\alpha) \sim n, f(\beta) \sim n, s(\beta) \leq \alpha \vdash \beta < \alpha \wedge f(\alpha) \sim f(\beta)$
$(C'5)$	$\bigvee_{i=0}^{n} f(\alpha) \sim i \vdash f(\alpha) \sim 0, \cdots, f(\alpha) \sim n$

Foundation for Natural-like Numbers

- An interesting pattern in the induction of the resolution refutation is the unifier σ always substitutes one variable in the term with the nested max term $m_n(n, \bar{x}_n)$ and renames the rest of the variables.
- Taking the standard interpretation of $m_n(n, \bar{x}_n)$ (after setting all variables to zero) we see that it is equivalent to 1.
- if one of the variables in a $m_n(n, \bar{x}_n)$ term is unified using σ we get a term equivalent to 2 in the standard interpretation (again setting variables to zero).
- We will define equivalence classes based on the nesting of the max functions.

Natural-like Numbers Example

Example

$$\overline{\mathbf{2}}_{(x,1)} = \left\{ \begin{array}{c} max(max(s(x_0), s(x_0)), s(max(max(s(x_0), s(x_0)), s(x_1)))) \\ \\ max(max(s(max(max(s(x_0), s(x_0)), s(x_1))), \\ s(max(max(s(x_0), s(x_0)), s(x_1)))), s(x_1)) \\ \\ max(max(s(max(max(s(x_0), s(x_0)), s(x_1))), \\ s(max(max(s(x_0), s(x_0)), s(x_1)))), \\ s(max(max(s(x_0), s(x_0)), s(x_1)))) \end{array} \right.$$

New Basecase for Well Ordering

- Using the extended clause set $C'(n)$ one can derive the following sequent from the proof of refutability where $0 \leq k \leq n$:

$$\bigwedge_{i=0}^{k} f(x_{b(i)}) \sim b(i), \bigvee_{i=0}^{n} f(\overline{\mathbf{1}}_{(x,n)}) \sim i \vdash$$

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

- Based on the anecdote about the term $m_n(n, \bar{x}_n)$ being a representation of 1, we can show that with the right constraints on the unifiers, we can do addition with the unifiers.

Resulting Herbrand Sequent

- Following the unifications through the induction steps of the resolution refutation results in the following sequent:

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

- This is essentially the pigeonhole principle if we consider each equivalence class as a natural number.

Removing Equivalence Classes

- We can consider the equivalence classes in the Herbrand sequent in a simpler form by replacing predicates of the form $P(\overline{\mathbf{n}}_{(x,k)})$ with $\bigvee_{t\in\overline{\mathbf{n}}_{(x,k)}} P(t)$ or $\bigwedge_{t\in\overline{\mathbf{n}}_{(x,k)}} P(t)$ on the left side.
- There are many spurious formulae in the Herbrand sequent we have constructed here, but the resulting sequent provides us with the propositional heart of the first-order proof.

Future Work

- The original goal of this work was to study the expressiveness of the CERES$_s$ method.
- As a result we see that even relatively simple schematic formal proofs are too complex for CERES$_s$ to express their cut free versions.
- Our goal is to find a fragment of the $\mathbf{LKS}_{\mathcal{E}}$ calculus such that proofs formalized in this fragment have schematic clause sets with refutations expressible within the SRRC.
- We are currently focusing on fragments of $\mathbf{LKS}_{\mathcal{E}}$ which only have monadic functions symbols in the term language.

Thank you for your time.

David Cerna.
Extracting schematic herbrand sequent from the analysis by ceress of a formal proof of nia.
.

David Cerna and Alexander Leitsch.
Analysis of a formal proof of the non-injectivity assertion using ceress.
.

Cvetan Dunchev, Alexander Leitsch, Mikheil Rukhaia, and Daniel Weller.
Ceres for first-order schemata, 2012.
http://arxiv.org/abs/1303.4257.