Integer Sequences, Recursive Lemma Elimination and Combinatorics

Vienna PhD School of Informatics

David M. Cerna
cernadavid1@logic.at

Vienna University of Technology
Institute of Computer Languages
Theory and Logic Group
Advisor: Prof. Alexander Leitsch

Lemma Elimination in Formal Proofs

What are Lemmata in Formal Proofs?

A lemma is an argument which shows up as an assumption and a consequent within the formal proof, i.e. non-essential. This can be written formally as follows:

$$\frac{\Gamma \vdash B, \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta', \Delta}\,\text{cut}$$

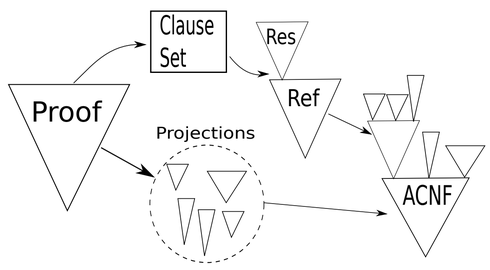
Where B would be the lemma and $\Gamma, \Gamma', \Delta', \Delta$ are context required for the final statement at the end of the proof. Lemmata are referred to as cuts in formal proofs.

Are all Cuts in a Formal Proofs Eliminable?

This property (cut-elimination) was shown for the *first-order LK calculus* by Gerhard Gentzen in 1934 by reducing the complexity of the formula representing the cut, i.e. break the cut into simpler cuts.

Cut-elimination by Resolution (CERES)

(Baaz et al. 2000) introduced another method of cut-elimination using resolution (J.A. Robinson 1965), a complete method for showing that a first order formula is unsatisfiable. By taking advantage of the fact that the conjunction of a formula used as an assumption and as a consequence is unsatisfiable we can use it for cut-elimination. A formal proof is then constructed using the resolution refutation as a backbone. Important to note is that any resolution refutation of the clause set derived from the lemmata of the proof can be used.



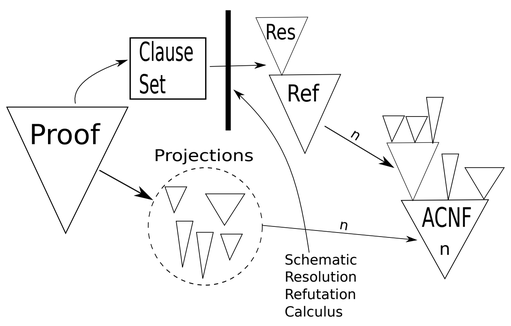
Recursive Formal Proofs

Consider, formal proofs which call themselves like primitive recursive functions, the concept is analogous to certain types of induction.

$$P(n+1) \Rightarrow Q\,[P(n)] \tag{1a}$$

$$P(0) \Rightarrow D \tag{1b}$$

Let D be a first-order logic proof, $Q\,[\cdot]$ is a first order logic proof containing at least one instance of the proof in the brackets, and n is a natural number. The Gentzen method of cut-elimination does not work when induction is present, however, replacing inductions with recursion (single parameter induction) allows application of the CERES method to the recursive proofs as was shown in (Dunchev et al. 2013).



However, resolution for such formulae is not well defined and a purpose built resolution method was needed, this method is neither sound nor complete.

Expressive Power Investigation

It was shown (Dunchev et al. 2013) that for recursive formal proofs whose clause set has a resolution refutation using the schematic resolution method have eliminable cuts, though a class of such proofs has not been accurately described. Thus, we used the method to analyse three proofs of varying complexity based on the *infinitary pigeon hole principle* in an attempt to find such a class of formal proofs. The statements used for these proofs are as follows:

- Let f be a total monotonically decreasing function over the natural numbers, then f is eventually constant.
- (NiA Schema): 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 $f(i) = f(j)$.
- (gNiA Schema): Let $f : \mathbb{N} \to [0, \cdots, n]$, where $n \in \mathbb{N}$, be total, then there exists $x_1, \cdots, x_m \in \mathbb{N}$ such that $x_1 < \cdots < x_m$ and $f(x_1) = f(x_2) = \cdots = f(x_m)$.

We will focus on the blue statement which has a proof using the following sequence of lemmata:

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* $x \leqslant y$ *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* $x \leqslant y$ *and* $f(y) = n + 1$.

Clause set for NiA Schema

$(C1)$	$\vdash \alpha \leqslant \alpha$
$(C2)$	$max(\alpha, \beta) \leqslant \gamma \vdash \alpha \leqslant \gamma$
$(C3)$	$max(\alpha, \beta) \leqslant \gamma \vdash \beta \leqslant \gamma$
$(C4_0)$	$f(\beta) \sim 0, f(\alpha) \sim 0, s(\beta) \leqslant \alpha \vdash$
$\vdots$	$\vdots$
$(C4_n)$	$f(\beta) \sim n, f(\alpha) \sim n, s(\beta) \leqslant \alpha \vdash$
$(C5)$	$\vdash f(\alpha) \sim 0, \cdots, f(\alpha) \sim n$

Clauses with length n(*schematic length clauses*) are important for analysis of the clause set. In this case, clause $(C5)$ is the only clause in the clause set with this property. Such clauses can be used to attain complexity results for the size of the final refutation.

A Derived Clause set for the NiA Schema

We used the SPASS theorem prover (http://www.spass-prover.org/) on instances of the clause set in an attempt to find essential derived clauses. It is not directly possible to make a schematic refutation from the SPASS output, because the theorem prover output is not recursive (no induction invariant is given).
Thus, we looked for ways to represent the patterns found in the output using bijective functions over $[0, \cdots, n]$ and we derived the following clause set (over all possible bijections b and all $-1 \leqslant k \leqslant n$) which is also refutable:

$$C_b(k) = \bigwedge_{i=0}^{k} f(x_{b(i)}) \sim b(i) \vdash \bigvee_{i=k+1}^{n} f(m(k+1, \bar{x}_{k+1}, z)) \sim b(i)$$

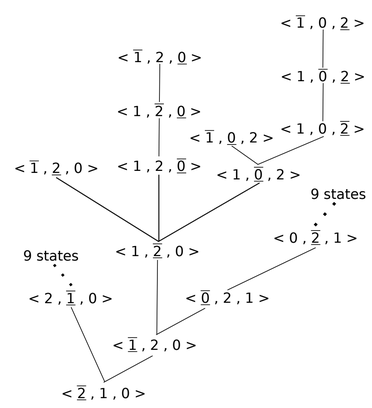
Fixing the value of k and evaluating only the antecedent, the number of clauses with that configuration is $\binom{n}{k}$. Further analysis provided a proof of refutability and a complexity result in terms of the number of times (C5) was used:

$$f(n) = n! \cdot \sum_{i=0}^{n} \frac{1}{i!} \tag{2}$$

This function can be found on OEIS as sequence (A000522). At this point we started to consider the combinatorial structure of the refutation in order to find a recursive language to express it (the language of (Dunchev et al. 2013) is not expressive enough) , namely, we considered the usage of permutations within the refutation.

Permutation Vectors and Resolution

- To express the refutation within a recursive language we need to provide a unique state to each use of the induction invariant.
- Iterating through all permutations would not provide enough states, thus we add labels to the permutations and rules for the interaction of the labels. We call the objects permutation vectors.



Starting at the permutation vector $\langle \overline{n}, n-1, \cdots, 0 \rangle$ and using a set of rewrite rules one can generate a number of permutation vectors larger than Eq. 2. Using permutation vectors we found a previously unknown recursion for sequence (A093964):

$$f(n) = n \cdot f(n-1) + \sum_{k=0}^{n-1} \frac{n!}{k!}$$

$$f(0) = 0$$

We can express Eq. 2 recursively using this combinatorial structure.

Future Work & gNiA Schema Clause set

$(C^g1) : \vdash \alpha \leqslant \alpha$

$(C^g2) : max(\alpha, \beta) \leqslant \gamma \vdash \alpha \leqslant \gamma$

$(C^g3) : max(\alpha, \beta) \leqslant \gamma \vdash \beta \leqslant \gamma$

$(C^g4_0) : \left(\bigwedge_{i=0}^{m+1} f(\alpha_i) \sim 0\right) \wedge \left(\bigwedge_{i=0}^{m} s(\alpha_i) \leqslant \alpha_{i+1}\right) \vdash$

$\vdots$

$(C^g4_n) : \left(\bigwedge_{i=0}^{m+1} f(\alpha_i) \sim n\right) \wedge \left(\bigwedge_{i=0}^{m} s(\alpha_i) \leqslant \alpha_{i+1}\right) \vdash$

$(C^g5) : \vdash \bigvee_{i=0}^{n} f(\alpha) \sim i$

In the case of the gNiA schema, we fixed n and left m free in our analysis. When we set $n = 2$, the growth rate of a part of the refutation is the sequence for the central binomial distribution (A000984). When $n = 3$ the same part of the refutation has a much more complex growth rate.
However, it can be expressed as the problem: Given the alphabet $\{a, b, c\}$ how many words can be constructed such that a shows up m times and b, c show up at most m times. We found a bijection between this problem and the permutations of sequence (A241193). It is still open whether we can construct a recursion using this combinatorial set.

Other than completing the analysis of the gNiA schema, we wish to investigate how general this method is, as in, can this be done and for a general class of schema. Also, we will investigate the possibility of using such a method for generating unknown integer sequences as well as the possibility of taking combinatorial structures with recursive definitions and finding proof schemata which require the structure.