Proof Analysis and Induction: Clausal Analysis of Proof Schemata

David M. Cerna and Michael Lettmann

September 10, 2017

JOHANNES KEPLER UNIVERSITY LINZ

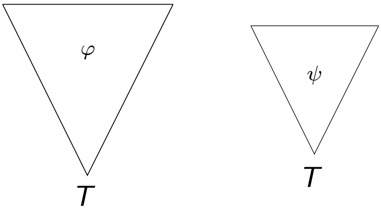
TU WIEN

Proof Analysis in a Nutshell

- Lets consider the following situation:

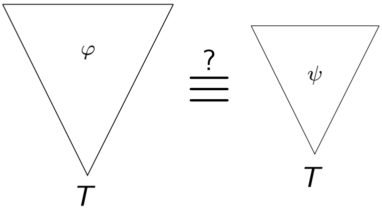
Proof Analysis in a Nutshell

- Lets consider the following situation:



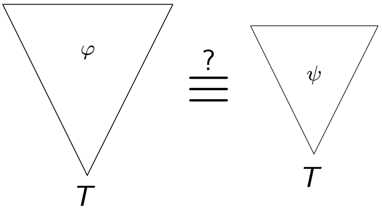
Proof Analysis in a Nutshell

- Lets consider the following situation:



Proof Analysis in a Nutshell

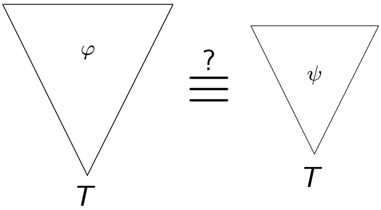
- Lets consider the following situation:



- Are φ and ψ essentially the same?
- Is the theory used to prove T necessary?
- What are the core principles necessary to prove T?
- Do φ and ψ share these core principles?
- Is there a unique set of necessary core principle?

Proof Analysis in a Nutshell

- Lets consider the following situation:



- Are φ and ψ essentially the same?
- Is the theory used to prove T necessary?
- What are the core principles necessary to prove T?
- Do φ and ψ share these core principles?
- Is there a unique set of necessary core principle?
- These questions are not completely trivial.

Proof Analysis and Induction

- Induction is one of the primary tools in the mathematicians tool box.

Proof Analysis and Induction

- Induction is one of the primary tools in the mathematicians tool box.
- Unfortunately induction adds additional obstacles to the already difficult task of proof analysis.
- Essentially, it is an abbreviation of “infinitary” argumentation.

Proof Analysis and Induction

- Induction is one of the primary tools in the mathematicians tool box.
- Unfortunately induction adds additional obstacles to the already difficult task of proof analysis.
- Essentially, it is an abbreviation of “infinitary” argumentation.

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\vdash P(0) \qquad \vdash P(0) \vdash P(1)}{\vdash P(1)}\,\text{cut}}{\vdots} \qquad P(t-1) \vdash P(t)}{\vdash P(t)} \qquad P(t) \vdash P(t+1)}{\vdash P(t+1)}\,\text{cut}}{\vdots}}{\vdash P(\beta)} \quad \Longrightarrow \quad \cfrac{\vdash P(0) \qquad P(\alpha) \vdash P(\alpha+1)}{\vdash P(\beta)}\,\text{IND}$$

Fürstenberg, primes, and an infinitude of proofs

Fürstenberg produced many proofs of elementary results using an unexpected intermediate theory.

- Local cut-elimination based proof analysis used by Jean-Yves Girard on Fürstenberg's proof of Van der Waerden's theorem.
- Proof analysis of his proof of the infinitude of primes was performed by Baaz *et al.* using a global cut-elimination procedure (**CERES**).

(Tangent) Global versus Local cut-elimination

Local cut-elimination reduces a cuts formula complexity or its distance from the leaves.

- Introduced by Gentzen as a method of proving consistency, the concept has been expanded well beyond the intended scope.

(Tangent) Global versus Local cut-elimination

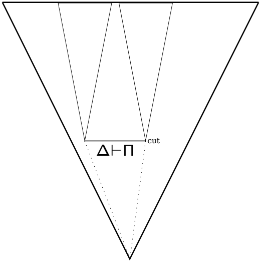
Local cut-elimination reduces a cuts formula complexity or its distance from the leaves.

- Introduced by Gentzen as a method of proving consistency, the concept has been expanded well beyond the intended scope.

Global cut-elimination produces an intermediate representation of a formal proofs cut-structure.

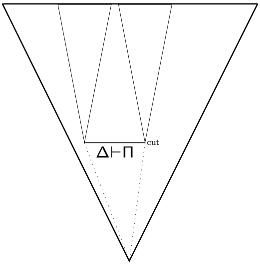
- From this intermediate representation a new proof with a **trivial cut-structure** is produced.

CERES: The Characteristic Clause Set representation

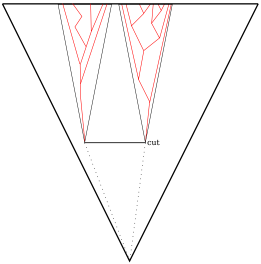


LK-Proof with cuts

CERES: The Characteristic Clause Set representation

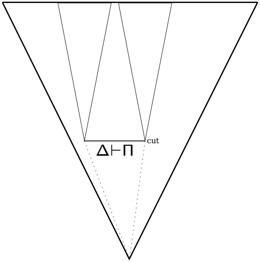


LK-Proof with cuts

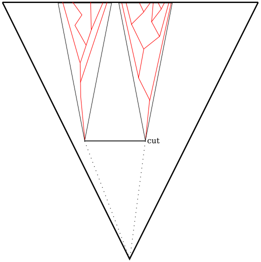


Paths to cut ancestors

CERES: The Characteristic Clause Set representation



LK-Proof with cuts

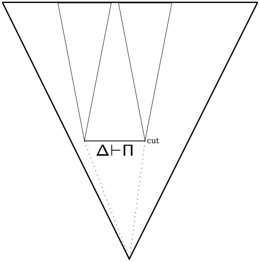


Paths to cut ancestors

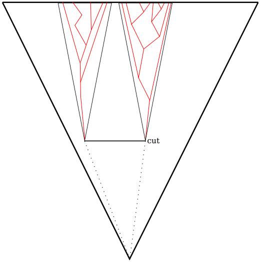
$$CL(A\vdash A)\equiv\{\vdash A\}$$
$$CL(A\vdash A)\equiv\{A\vdash\}$$
$$CL(A\vdash A)\equiv\{A\vdash A\}$$

- Construct a clause set from the cut ancestors relation.
- Such a clause set is always unsatisfiable.

CERES: The Characteristic Clause Set representation



LK-Proof with cuts



Paths to cut ancestors

$CL(A \vdash A) \equiv \{\vdash A\}$

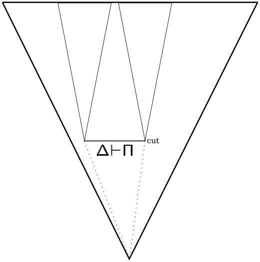
$CL(A \vdash A) \equiv \{A \vdash\}$

$CL(A \vdash A) \equiv \{A \vdash A\}$

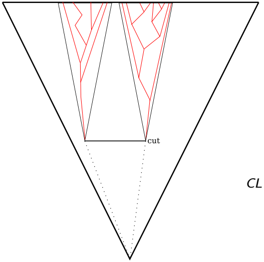
$$CL\left(\frac{\Delta \vdash \Pi}{\Delta' \vdash \Pi'}\ \rho\right) \equiv CL(\Delta \vdash \Pi)$$

- Construct a clause set from the cut ancestors relation.
- Such a clause set is always unsatisfiable.

CERES: The Characteristic Clause Set representation



LK-Proof with cuts



Paths to cut ancestors

$$CL(A \vdash A) \equiv \{\vdash A\}$$
$$CL(A \vdash A) \equiv \{A \vdash\}$$
$$CL(A \vdash A) \equiv \{A \vdash A\}$$

$$CL\left(\frac{\Delta \vdash \Pi}{\Delta' \vdash \Pi'}\ \rho\right) \equiv CL(\Delta \vdash \Pi)$$

$$CL\left(\frac{\Delta \vdash \Pi \qquad \Delta' \vdash \Pi'}{\Delta'' \vdash \Pi''}\ \rho\right) \equiv$$

$$\begin{cases} CL(\Delta \vdash \Pi) \cup CL(\Delta' \vdash \Pi') \\ CL(\Delta \vdash \Pi) \times CL(\Delta' \vdash \Pi') \end{cases}$$

- Construct a clause set from the cut ancestors relation.
- Such a clause set is always unsatisfiable.

Fürstenberg's proof as a Sequence of Proofs

Fürstenberg proof of the infinitude of primes is inductive.

- Unfortunately, local cut-elimination on inductive arguments is not a lossless procedure and in the worst case it's not possible.
- In [Baaz *et al.* 2008] proof analysis of Fürstenberg's proof was performed using a global cut-elimination by externalizing the inductive arguments.
- They formalized the proof as a **sequence** of cases, i.e assume n primes exists...
- The end result, a "schema" of proofs.

Fürstenberg's schema and its Clause Set

After schematizing the proof a componentwise characteristic clause set is be extracted.

- A refutation of each clause set was transformed into a proof skeleton upon which **projections** of the original proof are attached (**CERES**).
- Analysis of this schema of clause sets resulted in the discovery of Euclid's argument as well as other unknown combinatorial proofs within Fürstenberg's proof.

Fürstenberg's schema and its Clause Set

After schematizing the proof a componentwise characteristic clause set is be extracted.

- A refutation of each clause set was transformed into a proof skeleton upon which **projections** of the original proof are attached (**CERES**).
- Analysis of this schema of clause sets resulted in the discovery of Euclid's argument as well as other unknown combinatorial proofs within Fürstenberg's proof.

Later work [Dunchev *et al.* 2013] and [Leitsch *et al.* 2017] formalize the above procedure, while in [Cerna *et al.* 2016] the earlier method is used to perform proof analysis of a weak version of the pigeonhole principle.

Schematic **CERES**, and Fürstenberg's Proof

Analysis of Fürstenberg's Proof has not been performed using the method of [Dunchev *et al.* 2013] nor [Leitsch *et al.* 2017].

- Though neither method can deal with equational reasoning,
- the really problem is constructing a representable recursive refutation:

Schematic **CERES**, and Fürstenberg's Proof

Analysis of Fürstenberg's Proof has not been performed using the method of [Dunchev *et al.* 2013] nor [Leitsch *et al.* 2017].

- Though neither method can deal with equational reasoning,
- the really problem is constructing a representable recursive refutation:

$$\vdash E(f(x),0),\cdots E(f(x),\alpha+1) \qquad \vdash L(x,x)$$

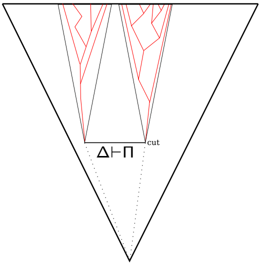
$$L(max(x,y),z)\vdash L(x,z) \qquad L(max(x,y),z)\vdash L(y,z)$$

$$E(f(x),0),E(f(y),0),L(s(y),x)\vdash \qquad E(f(x),1),E(f(y),1),L(s(y),x)\vdash$$

$$E(f(x),\alpha),E(f(y),\alpha),L(s(y),x)\vdash \qquad E(f(x),\alpha+1),E(f(y),\alpha+1),L(s(y),x)\vdash$$

Clausal analysis: Reductive C.E. and Global Cut Structure

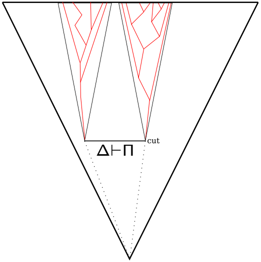
- Baaz and Leitsch, 2006 show how locally reducing cuts impacts the global cut structure.
- Every proof can be transformed into a proof with a minimally complex cut structure.
- The extracted clause set, is subsumed by the clause sets of the more complex cut structure.



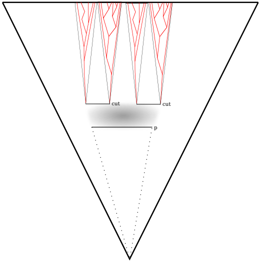
Reduction can result in

Clausal analysis: Reductive C.E. and Global Cut Structure

- Baaz and Leitsch, 2006 show how locally reducing cuts impacts the global cut structure.
- Every proof can be transformed into a proof with a minimally complex cut structure.
- The extracted clause set, is subsumed by the clause sets of the more complex cut structure.



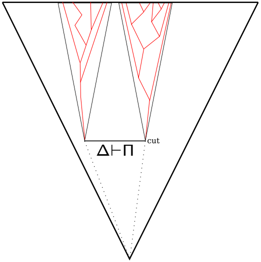
Reduction can result in



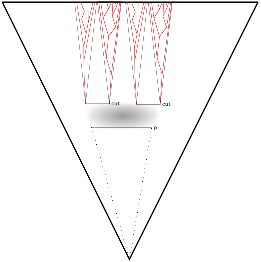
the following proof.

Clausal analysis: Reductive C.E. and Global Cut Structure

- Baaz and Leitsch, 2006 show how locally reducing cuts impacts the global cut structure.
- Every proof can be transformed into a proof with a minimally complex cut structure.
- The extracted clause set, is subsumed by the clause sets of the more complex cut structure.



Reduction can result in

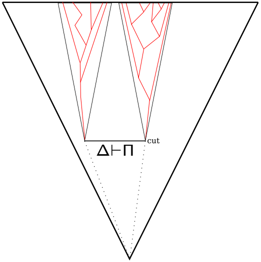


the following proof.

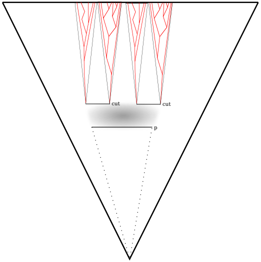
- Local elimination can result in a multiplication of the cuts
- Essentially, the cut-structure gets more redundant.

Clausal analysis: Reductive C.E. and Global Cut Structure

- Baaz and Leitsch, 2006 show how locally reducing cuts impacts the global cut structure.
- Every proof can be transformed into a proof with a minimally complex cut structure.
- The extracted clause set, is subsumed by the clause sets of the more complex cut structure.



Reduction can result in



the following proof.

- Local elimination can result in a multiplication of the cuts
- Essentially, the cut-structure gets more redundant.
- Redundancy $\equiv$ **structural simplicity**, refuting is easier.

The Structurally Simplest Clause set

- The size of the **top clause set** is exponential in the number unique literals.
- Consider the following:

The Structurally Simplest Clause set

- The size of the **top clause set** is exponential in the number unique literals.
- Consider the following:

Clause Set

$$\begin{array}{r} \vdash P(1) \\ \vdash P(2) \\ P(2) \vdash P(3) \\ P(2) \vdash P(4) \\ P(1), Q(2), Q(1) \vdash \\ P(1), R(2), R(1) \vdash \\ P(4), Q(3), Q(1) \vdash \\ P(4), R(3), R(1) \vdash \\ P(3), Q(3), Q(2) \vdash \\ P(3), R(3), R(2) \vdash \\ \vdash Q(1), R(1) \\ \vdash Q(2), R(2) \\ \vdash Q(3), R(3) \end{array}$$

Top Clause Set

$$\begin{array}{r} Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3), P(3), P(4) \vdash \\ Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3), P(3) \vdash P(4) \\ Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3), P(4) \vdash P(3) \\ Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3) \vdash P(3), P(4) \\ Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), P(3), P(4) \vdash R(3) \\ Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), P(3) \vdash R(3), P(4) \\ \vdots \\ R(3), P(4) \vdash Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), P(3) \\ R(3) \vdash Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), P(3), P(4) \\ P(3), P(4) \vdash Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3) \\ P(3) \vdash Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3), P(4) \\ P(4) \vdash Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3), P(3) \\ \vdash Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3), P(3), P(4) \end{array}$$

The Structurally Simplest Clause set

- The size of the **top clause set** is exponential in the number unique literals.
- Consider the following:

Clause Set

$$\vdash P(1)$$
$$\vdash P(2)$$
$$P(2) \vdash P(3)$$
$$P(2) \vdash P(4)$$
$$P(1), Q(2), Q(1) \vdash$$
$$P(1), R(2), R(1) \vdash$$
$$P(4), Q(3), Q(1) \vdash$$
$$P(4), R(3), R(1) \vdash$$
$$P(3), Q(3), Q(2) \vdash$$
$$P(3), R(3), R(2) \vdash$$
$$\vdash Q(1), R(1)$$
$$\vdash Q(2), R(2)$$
$$\vdash Q(3), R(3)$$

Top Clause Set

$$Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3), P(3), P(4) \vdash$$
$$Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3), P(3) \vdash P(4)$$
$$Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3), P(4) \vdash P(3)$$
$$Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3) \vdash P(3), P(4)$$
$$Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), P(3), P(4) \vdash R(3)$$
$$Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), P(3) \vdash R(3), P(4)$$
$$\vdots$$
$$R(3), P(4) \vdash Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), P(3)$$
$$R(3) \vdash Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), P(3), P(4)$$
$$P(3), P(4) \vdash Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3)$$
$$P(3) \vdash Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3), P(4)$$
$$P(4) \vdash Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3), P(3)$$
$$\vdash Q(1), R(1), P(1), Q(2), R(2), P(2), Q(3), R(3), P(3), P(4)$$

HAS 1024 CLAUSES!!!

Refutations of Top Clause Sets

- Top clause sets are huge but easy to refute.
- There is pretty much only one way to refute them.

$$\dfrac{\dfrac{\vdash \Delta, P(3), P(4) \vdash \qquad \vdash \Delta, P(3) \vdash P(4)}{\Delta, P(3) \vdash} \qquad \dfrac{\vdash \Delta, R(3), P(4) \vdash P(3) \qquad \vdash \Delta \vdash P(3), P(4)}{\Delta, R(3) \vdash P(3)}}{\Delta, R(3) \vdash}$$

$$\vdots$$

- As one might imagine to refute $\Delta, R(3) \vdash$ we need a derivation using

$$\Delta \vdash R(3), P(3), P(4)$$
$$\Delta \vdash R(3), P(3)$$

- similar to the construction of a semantic tree.

Benefits of Top Clause Sets

- The structurally simplicity of top clause set allow a compact representation of their refutation.
- [Condoluci, 2016], for propositional logic, showed that an ordered sequence of the atoms $\mathcal{A}$ can produce a top clause set with the following compact refutation:

$$REF(\{P\} \cup \mathcal{A}', X) = \quad \frac{REF(\mathcal{A}', X \circ \vdash P) \qquad REF(\mathcal{A}', X \circ P \vdash)}{X} \; Res$$

$$REF(\emptyset, X) = X$$

- Top clause sets and the above results provide a **cut-elimination complete method** for *Schematic* Propositional Logic.

Schematic Proofs

- Before further discussing schematic clausal analysis and top clause sets we introduce proof schemata:

Schematic Proofs

- Before further discussing schematic clausal analysis and top clause sets we introduce proof schemata:

$$\frac{\Sigma \vdash P(0), \Delta \qquad \Pi, P(\alpha) \vdash P(s(\alpha)), \Gamma}{\Pi, \Sigma \vdash P(\beta), \Delta, \Gamma}$$

Schematic Proofs

- Before further discussing schematic clausal analysis and top clause sets we introduce proof schemata:

$$\frac{\Sigma \vdash P(0), \Delta \qquad \Pi, P(\alpha) \vdash P(s(\alpha)), \Gamma}{\Pi, \Sigma \vdash \underset{P(s(\alpha))}{\cancel{P(\beta)}}, \Delta, \Gamma}$$

Schematic Proofs

- Before further discussing schematic clausal analysis and top clause sets we introduce proof schemata:

$$
\dfrac{\dfrac{\dfrac{\Psi}{\Pi' \vdash \Gamma'}}{\vdots} \qquad \dfrac{\dfrac{\varphi(0)}{\Sigma \vdash P(0), \Delta}}{\vdots}}{\Pi'', \Sigma \vdash P(1), \Delta, \Gamma''}
$$

$$
\dfrac{\Sigma \vdash P(0), \Delta \qquad \Pi, P(\alpha) \vdash P(s(\alpha)), \Gamma}{\Pi, \Sigma \vdash \cancel{P(\beta)}, \Delta, \Gamma} \qquad \Rightarrow
$$

$$
P(s(\alpha))
$$

Schematic Proofs

- Before further discussing schematic clausal analysis and top clause sets we introduce proof schemata:

$$
\begin{array}{c}
\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\overset{\varphi(0)}{\cdots\cdots\cdots}}{\Sigma \vdash P(0), \Delta} \\
\vdots \qquad\qquad \vdots \\
\hline
\Pi'', \Sigma \vdash P(1), \Delta, \Gamma''
\end{array}
$$

$$\Downarrow$$

$$
\dfrac{\Sigma \vdash P(0), \Delta \qquad \Pi, P(\alpha) \vdash P(s(\alpha)), \Gamma}{\Pi, \Sigma \vdash \underset{P(s(\alpha))}{\cancel{P(\beta)}}, \Delta, \Gamma}
\quad \Rightarrow \quad
\begin{array}{c}
\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\overset{\varphi(1)}{\cdots\cdots\cdots\cdots\cdots}}{\Pi'', \Sigma \vdash P(1), \Delta, \Gamma''} \\
\vdots \qquad\qquad \vdots \\
\hline
\Pi''', \Sigma \vdash P(2), \Delta, \Gamma'''
\end{array}
$$

Schematic Proofs

- Before further discussing schematic clausal analysis and top clause sets we introduce proof schemata:

$$
\dfrac{\Sigma \vdash P(0), \Delta \qquad \Pi, P(\alpha) \vdash P(s(\alpha)), \Gamma}{\Pi, \Sigma \vdash \underset{P(s(\alpha))}{\cancel{P(\beta)}}, \Delta, \Gamma}
\quad \Rightarrow \quad
$$

$$
\dfrac{\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\overset{\varphi(0)}{\cdots\cdots}}{\Sigma \vdash P(0), \Delta}}{\vdots \qquad\qquad \vdots}
$$

$$
\Pi'', \Sigma \vdash P(1), \Delta, \Gamma''
$$

$$\Downarrow$$

$$
\dfrac{\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\overset{\varphi(1)}{\cdots\cdots}}{\Pi'', \Sigma \vdash P(1), \Delta, \Gamma''}}{\vdots \qquad\qquad \vdots}
$$

$$
\Pi''', \Sigma \vdash P(2), \Delta, \Gamma'''
$$

$$\Downarrow \quad \alpha - 2 \ \textit{times}$$

$$
\dfrac{\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\overset{\varphi(\alpha)}{\cdots\cdots}}{\Pi^{(\alpha+1)}, \Sigma \vdash P(\alpha), \Delta, \Gamma^{(\alpha+1)}}}{\vdots \qquad\qquad \vdots}
$$

$$
\Pi, \Sigma \vdash P(\alpha + 1), \Delta, \Gamma
$$

Schematic Proofs

- Before further discussing schematic clausal analysis and top clause sets we introduce proof schemata:

$$\dfrac{\Sigma \vdash P(0), \Delta \qquad \Pi, P(\alpha) \vdash P(s(\alpha)), \Gamma}{\Pi, \Sigma \vdash \cancel{P(\beta)}, \Delta, \Gamma}$$
$$P(s(\alpha))$$

$$\Rightarrow$$

$$\dfrac{\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\overset{\varphi(0)}{\cdots\cdots}}{\Sigma \vdash P(0), \Delta}}{\dfrac{\vdots \qquad\qquad \vdots}{\Pi'', \Sigma \vdash P(1), \Delta, \Gamma''}}$$

$$\Downarrow$$

$$\dfrac{\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\overset{\varphi(1)}{\cdots\cdots}}{\Pi'', \Sigma \vdash P(1), \Delta, \Gamma''}}{\dfrac{\vdots \qquad\qquad \vdots}{\Pi''', \Sigma \vdash P(2), \Delta, \Gamma'''}}$$

$$\Downarrow \quad \alpha - 2 \ \textit{times}$$

$$\dfrac{\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\overset{\varphi(\alpha)}{\cdots\cdots}}{\Pi^{(\alpha+1)}, \Sigma \vdash P(\alpha), \Delta, \Gamma^{(\alpha+1)}}}{\dfrac{\vdots \qquad\qquad \vdots}{\Pi, \Sigma \vdash P(\alpha+1), \Delta, \Gamma}}$$

- The Proof is indexed by α.
- Instantiating α results in an **LK**-proof **.**
- Proof anaylsis without instantiation.

Example: Proof Schema

- Let $\Phi = \langle(\varphi, \pi, \nu(k))\rangle$ and
  $\mathcal{E} = \left\{ \hat{S}(k+1) = s(\hat{S}(k)) \;;\; \hat{S}(0) = 0 \;;\; k + s(l) = s(k+l) \right\}$.

$$\pi = \dfrac{\dfrac{P(\alpha+0) \vdash P(\alpha+0)}{P(\alpha+0), \forall x.P(x) \to P(s(x)) \vdash P(\alpha+0)}\, w\colon l}{P(\alpha+0), \forall x.P(x) \to P(s(x)) \vdash P(\alpha+\hat{S}(0))}\, \mathcal{E}$$

$$\nu(k) = \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\varphi, n, \alpha)}{P(\alpha+0), \forall x.P(x) \to P(s(x)) \vdash P(\alpha+\hat{S}(n))} \qquad P(s(\alpha+\hat{S}(n))) \vdash P(s(\alpha+\hat{S}(n)))}{P(\alpha+0), \forall x.P(x) \to P(s(x)), P(\alpha+\hat{S}(n)) \to P(s(\alpha+\hat{S}(n))) \vdash P(s(\alpha+\hat{S}(n)))}\, \to\colon l}{P(\alpha+0), \forall x.P(x) \to P(s(x)), \forall x.P(x) \to P(s(x)) \vdash P(s(\alpha+\hat{S}(n)))}\, \forall\colon l}{P(\alpha+0), \forall x.P(x) \to P(s(x)), \forall x.P(x) \to P(s(x)) \vdash P(\alpha+s(\hat{S}(n)))}\, \mathcal{E}}{P(\alpha+0), \forall x.P(x) \to P(s(x)), \forall x.P(x) \to P(s(x)) \vdash P(\alpha+\hat{S}(n+1))}\, \mathcal{E}}{P(\alpha+0), \forall x.P(x) \to P(s(x)) \vdash P(\alpha+\hat{S}(n+1))}\, c\colon l$$

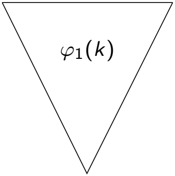
Why [Baaz and Leitsch, 2006] is not Enough

- In [Baaz and Leitsch, 2006], local cut-elimination is used to perform clausal analysis.
- Unfortunately, this method fails when a cut reduction step reaches a link.

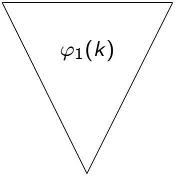
$$\dfrac{\dfrac{(\varphi_I, t, \bar{x})}{C, \Delta \vdash \Gamma} \qquad \dfrac{(\varphi_j, t', \bar{x})}{\Delta' \vdash \Gamma', C}}{\Delta, \Delta' \vdash \Gamma, \Gamma'}\,\text{cut}$$

- To solve the problem we need to see proof schemata as more than a recursive **LK**-proof.
- Extend local cut-elimination to proof schemata.

Towards Schematic Local Cut-Elimination



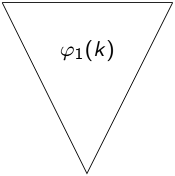
$\Phi = \langle \varphi_1, \cdots, \varphi_n \rangle$



$$\Phi = \langle {\color{red}\varphi_1}, \cdots, \varphi_n \rangle$$

$$\varphi_1.s = \frac{(\varphi_2, k, \bar{t})}{S(k, \bar{t})}$$

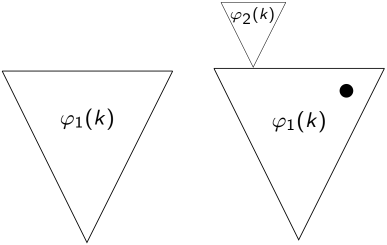
Towards Schematic Local Cut-Elimination



$$\Phi = \langle \varphi_1, \cdots, \varphi_n \rangle$$

$$\varphi_1.s = \quad \frac{(\varphi_2, k, \bar{t})}{S(k, \bar{t})} \qquad\qquad \varphi_1.b = \varphi_2.b$$

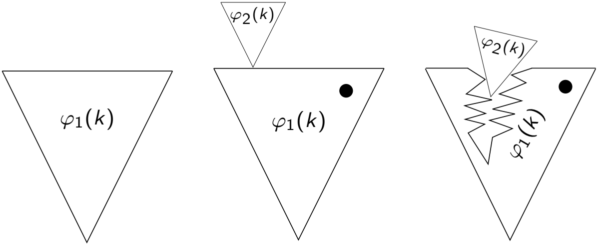
Towards Schematic Local Cut-Elimination



$\Phi = \langle \varphi_1, \cdots, \varphi_n \rangle$ "*Lazy* Instantiation"

$$\varphi_1.s = \frac{(\varphi_2, k, \bar{t})}{S(k, \bar{t})} \qquad \varphi_1.b = \varphi_2.b$$

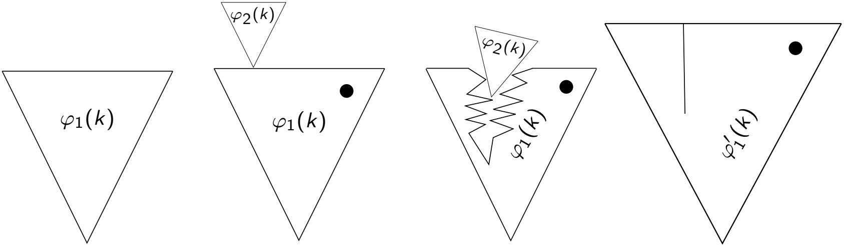
Towards Schematic Local Cut-Elimination



$\Phi = \langle \varphi_1, \cdots, \varphi_n \rangle$ "*Lazy* Instantiation" "THE ACCUMULATOR"

$$\varphi_1.s = \frac{(\varphi_2, k, \bar{t})}{S(k, \bar{t})} \qquad \varphi_1.b = \varphi_2.b$$

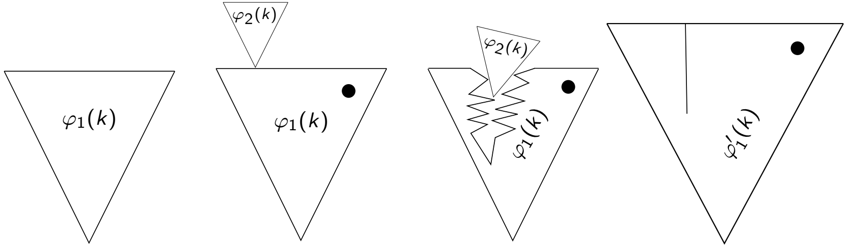
Towards Schematic Local Cut-Elimination



$\Phi = \langle \varphi_1, \cdots, \varphi_n \rangle$ "*Lazy* Instantiation" "THE ACCUMULATOR" $\Phi' = \langle \varphi'_1, \cdots, \varphi_n \rangle$

$$\varphi_1.s = \frac{(\varphi_2, k, \bar{t})}{S(k, \bar{t})} \qquad \varphi_1.b = \varphi_2.b$$

Towards Schematic Local Cut-Elimination



$\Phi = \langle \varphi_1, \cdots, \varphi_n \rangle$ "*Lazy* Instantiation" "THE ACCUMULATOR" $\Phi' = \langle \varphi_1', \cdots, \varphi_n \rangle$

$$\varphi_1.s = \frac{(\varphi_2, k, \bar{t})}{S(k, \bar{t})} \qquad \varphi_1.b = \varphi_2.b$$

$$\varphi_1'.b = \varphi_2.b$$
$$\varphi_1'.s = \varphi_2.s$$

Example: Lazy Instantiation

- Let $\Phi = \langle(\psi, \pi, \nu(k))\rangle$

$\nu(k) =$

$$
\dfrac{
  \dfrac{
    \dfrac{
      \varphi_{(n+1)}
      \qquad
      \dfrac{(\psi, n, \_)}{\bigwedge_{i=0}^{n} P(i) \rightarrow Q(i) \vdash \bigwedge_{i=0}^{n} \neg P(i) \vee Q(i)}\;\dashrightarrow
    }{
      \bigwedge_{i=0}^{n+1} P(i) \rightarrow Q(i), P(n+1) \rightarrow Q(n+1) \vdash \left(\bigwedge_{i=0}^{n} \neg P(i) \vee Q(i)\right) \wedge (\neg P(n+1) \vee Q(n+1))
    }\;\wedge : r
  }{
    \bigwedge_{i=0}^{n+1} P(i) \rightarrow Q(i), P(n+1) \rightarrow Q(n+1) \vdash \bigwedge_{i=0}^{n+1} \neg P(i) \vee Q(i)
  }\;\mathcal{E}
}{
  \dfrac{
    \left(\bigwedge_{i=0}^{n} P(i) \rightarrow Q(i)\right) \wedge P(n+1) \rightarrow Q(n+1) \vdash \bigwedge_{i=0}^{n+1} \neg P(i) \vee Q(i)
  }{
    \bigwedge_{i=0}^{n+1} P(i) \rightarrow Q(i) \vdash \bigwedge_{i=0}^{n+1} \neg P(i) \vee Q(i)
  }\;\mathcal{E}
}\;\wedge : l
$$

$$
\pi = \dfrac{
  \dfrac{
    \dfrac{P(0) \vdash P(0)}{\vdash \neg P(0), P(0)}\;\neg : r
    \qquad
    Q(0) \vdash Q(0)
  }{
    P(0) \rightarrow Q(0) \vdash \neg P(0), Q(0)
  }\;\rightarrow : l
}{
  P(0) \rightarrow Q(0) \vdash \neg P(0) \vee Q(0)
}\;\vee : r
$$

Lazy Instantiation by Example

- We construct $\Phi^a = \langle A, \mathbf{C}_1 \rangle$ from Φ by adding $A = (\chi, \pi', \nu'(k))$:

$\nu'(k) =$

$$\frac{(\psi, n+1, \_)}{\bigwedge_{i=0}^{n+1} P(i) \to Q(i) \vdash \bigwedge_{i=0}^{n+1} \neg P(i) \vee Q(i)}$$

$$\pi' = \dfrac{\dfrac{\dfrac{P(0) \vdash P(0)}{\vdash \neg P(0), P(0)}\ \neg : r \qquad Q(0) \vdash Q(0)}{P(0) \to Q(0) \vdash \neg P(0), Q(0)}\ \to : l}{P(0) \to Q(0) \vdash \neg P(0) \vee Q(0)}\ \vee : r$$

Lazy Instantiation by Example

- Instantiating $\nu'(k)$ with $\nu(n+1)$ we get a proof schema $\Phi^{a'} = \langle A', \mathbf{C}_1 \rangle$ from Φ by adding the component $A = (\chi, \pi', \hat{\nu}(k))$:

$$\hat{\nu}(k) = \cfrac{\cfrac{\cfrac{\cfrac{\varphi_{(n+1)} \qquad \cfrac{(\psi, n, \_)}{\bigwedge_{i=0}^{n} P(i) \to Q(i) \vdash \bigwedge_{i=0}^{n} \neg P(i) \lor Q(i)}}{\bigwedge_{i=0}^{n+1} P(i) \to Q(i), P(n+1) \to Q(n+1) \vdash (\bigwedge_{i=0}^{n} \neg P(i) \lor Q(i)) \land (\neg P(n+1) \lor Q(n+1))} \land : r}{\bigwedge_{i=0}^{n+1} P(i) \to Q(i), P(n+1) \to Q(n+1) \vdash \bigwedge_{i=0}^{n+1} \neg P(i) \lor Q(i)} \mathcal{E}}{(\bigwedge_{i=0}^{n} P(i) \to Q(i)) \land P(n+1) \to Q(n+1) \vdash \bigwedge_{i=0}^{n+1} \neg P(i) \lor Q(i)} \land : l}{\bigwedge_{i=0}^{n+1} P(i) \to Q(i) \vdash \bigwedge_{i=0}^{n+1} \neg P(i) \lor Q(i)} \mathcal{E}$$

- Ok nothing new yet, lets do it again

Lazy Instantiation by Example

- Repeating the process with get $\Phi^{a*} = \left\langle \hat{A}, \mathbf{C}_1 \right\rangle$ where $A = (\chi, \pi', \nu^*(k))$:

$\nu^*(k)=$

$$
\dfrac{
\dfrac{
\dfrac{
\varphi_{(n+2)} \qquad
\dfrac{
\dfrac{
\varphi_{(n+1)} \qquad
\dfrac{(\psi, n, \_)}{\bigwedge_{i=0}^{n} P(i) \to Q(i) \vdash \bigwedge_{i=0}^{n} \neg P(i) \lor Q(i)}
}{\vdots}\ \land : r
}{\bigwedge_{i=0}^{n+1} P(i) \to Q(i) \vdash \bigwedge_{i=0}^{n+1} \neg P(i) \lor Q(i)}\ c : l
}{\bigwedge_{i=0}^{n+1} P(i) \to Q(i), P() \to Q(n+2) \vdash \left(\bigwedge_{i=0}^{n+1} \neg P(i) \lor Q(i)\right) \land (\neg P(n+2) \lor Q(n+2))}\ \land : r
}{\vdots}\ \mathcal{E}
}{\bigwedge_{i=0}^{n+2} P(i) \to Q(i) \vdash \bigwedge_{i=0}^{n+2} \neg P(i) \lor Q(i)}\ c : l
$$

Map of results

$$
\begin{tikzcd}
\Phi \arrow[r, dashed, "\text{(i)}"] & \Phi^a \arrow[r, dashed, "\text{(ii)}"] \arrow[d, "\text{(iii)}"'] & \Phi' \arrow[rr, dashed, "\text{(v)}"] \arrow[d, "\text{(iii)}"] & & \Psi \arrow[d, "\text{(vi)}"] \arrow[ddll, "\text{(iii)}"'] & & \\
& \Phi^a \downarrow_\alpha \arrow[dr, "\text{(v)}"'] & \Phi' \downarrow_\beta \arrow[l, dashed, "\text{(iv)}"'] \arrow[d, "\text{(v)}"'] & & \Theta^\Psi \arrow[r, "\text{(x)}"] \arrow[d, "\text{(iii)}"] & \mathcal{A}^\Psi \arrow[dd, "\text{(xi)}"] & \\
& & \psi \arrow[r, "\text{(vi)}"'] \arrow[d, "\text{(v)}"'] & \Theta(\Psi \downarrow_\beta) \arrow[r, leftrightarrow, "\text{(viii)}"] \arrow[d, "\text{(vii)}"'] & \Theta^\Psi \downarrow_\beta \arrow[dl, dashed, "\text{(ix)}"] \arrow[dr, dashed, "\text{(ix)}"'] & & \\
& & \psi^{top} \arrow[r, "\text{(vi)}"'] & \Theta(\psi^{top}) \arrow[rr, dashed, "\text{(ix)}"] & & CL^t(\mathcal{A}^\Psi) \arrow[r, "\text{(xii)}"] & \vdash
\end{tikzcd}
$$

Lazy Instantiation and Lemma (1)

$$
\begin{tikzcd}
\Phi \arrow[r, dashed, "\text{(i)}"] & \Phi^a \arrow[r, dashed, "\text{(ii)}"] \arrow[d, "\text{(iii)}"'] & \Phi' \arrow[rr, dashed, "\text{(v)}"] \arrow[d, "\text{(iii)}"] & & \Psi \arrow[d, "\text{(vi)}"] \arrow[ddll, "\text{(iii)}"'] & & \\
& \Phi^a \downarrow_\alpha \arrow[dr, "\text{(v)}"'] & \Phi' \downarrow_\beta \arrow[l, dashed, "\text{(iv)}"'] \arrow[d, "\text{(v)}"'] & & \Theta^\Psi \arrow[r, "\text{(x)}"] \arrow[d, "\text{(iii)}"] & \mathcal{A}^\Psi \arrow[dd, "\text{(xi)}"] & \\
& & \psi \arrow[r, "\text{(vi)}"'] \arrow[d, "\text{(v)}"'] & \Theta(\Psi \downarrow_\beta) \arrow[r, leftrightarrow, "\text{(viii)}"] \arrow[d, "\text{(vii)}"'] & \Theta^\Psi \downarrow_\beta \arrow[dl, dashed, "\text{(ix)}"] \arrow[dr, dashed, "\text{(ix)}"'] & & \\
& & \psi^{top} \arrow[r, "\text{(vi)}"'] & \Theta(\psi^{top}) \arrow[rr, dashed, "\text{(ix)}"] & & CL^t(\mathcal{A}^\Psi) \arrow[r, "\text{(xii)}"] & \vdash
\end{tikzcd}
$$

Reductive Cut-Elimination and Lemma (2)

$$
\begin{tikzcd}
\Phi \arrow[r, dashed, "\text{(i)}"] & \Phi^a \arrow[r, dashed, "\text{(ii)}"] \arrow[dd, "\text{(iii)}"'] & \Phi' \arrow[rr, dashed, red, "\text{(v)}"] \arrow[dd, "\text{(iii)}"] & & \Psi \arrow[d, "\text{(vi)}"] \arrow[dddll, red, "\text{(iii)}"'] & & \\
& & & & \Theta^{\Psi} \arrow[r, "\text{(x)}"] \arrow[dd, "\text{(iii)}"] & \mathcal{A}^{\Psi} \arrow[ddd, "\text{(xi)}"] & \\
& \Phi^a \downarrow_{\alpha} \arrow[dr, red, "\text{(v)}"'] & \Phi' \downarrow_{\beta} \arrow[l, dashed, "\text{(iv)}"'] \arrow[d, red, "\text{(v)}"'] & & & & \\
& & \psi \arrow[r, "\text{(vi)}"'] \arrow[d, red, "\text{(v)}"'] & \Theta(\Psi \downarrow_{\beta}) \arrow[r, leftrightarrow, "\text{(viii)}"] \arrow[d, "\text{(vii)}"'] & \Theta^{\Psi} \downarrow_{\beta} \arrow[dl, dashed, "\text{(ix)}"] \arrow[dr, dashed, "\text{(ix)}"'] & & \\
& & \psi^{top} \arrow[r, "\text{(vi)}"'] & \Theta(\psi^{top}) \arrow[rr, dashed, "\text{(ix)}"] & & CL^t(\mathcal{A}^{\Psi}) \arrow[r, "\text{(xii)}"] & \vdash
\end{tikzcd}
$$

$$
\begin{tikzcd}
\Phi \arrow[r, dashed, "\text{(i)}"] & \Phi^a \arrow[r, dashed, "\text{(ii)}"] \arrow[d, "\text{(iii)}"'] & \Phi' \arrow[rr, dashed, "\text{(v)}"] \arrow[d, "\text{(iii)}"] & & \Psi \arrow[d, "\text{(vi)}"] \arrow[ddll, "\text{(iii)}"'] & & \\
& \Phi^a \downarrow_\alpha \arrow[dr, "\text{(v)}"'] & \Phi' \downarrow_\beta \arrow[l, dashed, "\text{(iv)}"'] \arrow[d, "\text{(v)}"'] & & \Theta^\Psi \arrow[r, "\text{(x)}"] \arrow[d, "\text{(iii)}"] & \mathcal{A}^\Psi \arrow[dd, "\text{(xi)}"] & \\
& & \psi \arrow[r, "\text{(vi)}"'] \arrow[d, "\text{(v)}"'] & \Theta(\Psi \downarrow_\beta) \arrow[r, leftrightarrow, "\text{(viii)}"] \arrow[d, "\text{(vii)}"'] & \Theta^\Psi \downarrow_\beta \arrow[dl, dashed, "\text{(ix)}"] \arrow[dr, dashed, "\text{(ix)}"'] & & \\
& & \psi^{top} \arrow[r, "\text{(vi)}"'] & \Theta(\psi^{top}) \arrow[rr, dashed, "\text{(ix)}"] & & CL^t(\mathcal{A}^\Psi) \arrow[r, "\text{(xii)}"] & \vdash
\end{tikzcd}
$$

Consequences and Future Work

$$
\begin{tikzcd}
\Phi \arrow[r, dashed, "(i)"] & \Phi^a \arrow[r, dashed, "(ii)"] \arrow[dd, "(iii)"'] & \Phi' \arrow[rr, dashed, "(v)"] \arrow[dd, "(iii)"] & & \Psi \arrow[d, "(vi)"] \arrow[dddll, "(iii)"'] & & \\
 & & & & \Theta^{\Psi} \arrow[r, "(x)"] \arrow[dd, "(iii)"] & \mathcal{A}^{\Psi} \arrow[ddd, "(xi)"] & \\
 & \Phi^a \downarrow_{\alpha} \arrow[dr, "(v)"'] & \Phi' \downarrow_{\beta} \arrow[l, dashed, "(iv)"'] \arrow[d, "(v)"'] & & & & \\
 & & \psi \arrow[r, "(vi)"'] \arrow[d, "(v)"'] & \Theta(\Psi \downarrow_{\beta}) \arrow[r, leftrightarrow, "(viii)"] \arrow[d, "(vii)"'] & \Theta^{\Psi} \downarrow_{\beta} \arrow[dl, dashed, "(ix)"] \arrow[dr, dashed, "(ix)"'] & & \\
 & & \psi^{top} \arrow[r, "(vi)"'] & \Theta(\psi^{top}) \arrow[rr, dashed, "(ix)"] & & CL^t(\mathcal{A}^{\Psi}) \arrow[r, "(xii)"] & \vdash
\end{tikzcd}
$$

Conclusion & Future Work

- The work presented here justifies an alternative method for dealing with the cut-structure of proof schemata.
- Essentially the results justify the procedure of [Condoluci, 2016] for propositional proof schemata.
- As for future work, we would like to extend the procedure of [Condoluci, 2016] to first order and investigate how to deal with substitutions in recursive resolution refutations .
- Also of interest is a generalization of lazy instantiation to any component in a proof schema.

Thank you for your time.