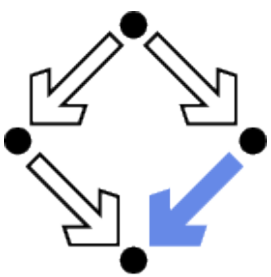
# Proof Schemata: A Formalism for Proof Transformation in the Presence of Induction

David M. Cerna

September 6$^{\text{th}}$, 2018

JOHANNES KEPLER
UNIVERSITY LINZ



# Transforming Sequences of Proofs

- A sequences of proofs was used to analyze Fürstenberg's proof of the infinitude of primes [Baaz et al. 2008].
- Each proof in the schema assumed a finite number of primes exists.
- A uniform structure indexed by the number of assumed primes was presented and used for the analysis.
- Proof Schema are a formal description of this representation.
- The uniform structure is a set of proofs containing Links, non-tautological axioms.
- Certain restrictions on links guarantee sound construction of proof schema

# Proof Schema as an Interpretation of Induction

- While initial work considered only a fragment of arithmetic, Schematic construction for Peano arithmetic are possible [ Cerna & Lolic , 2018].

# Proof Schema as an Interpretation of Induction

- While initial work considered only a fragment of arithmetic, Schematic construction for Peano arithmetic are possible [ Cerna & Lolic , 2018].

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

# Proof Schema as an Interpretation of Induction

- While initial work considered only a fragment of arithmetic, Schematic construction for Peano arithmetic are possible [ Cerna & Lolic , 2018].

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

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

# Proof Schema as an Interpretation of Induction

- While initial work considered only a fragment of arithmetic, Schematic construction for Peano arithmetic are possible [ Cerna & Lolic , 2018].

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

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

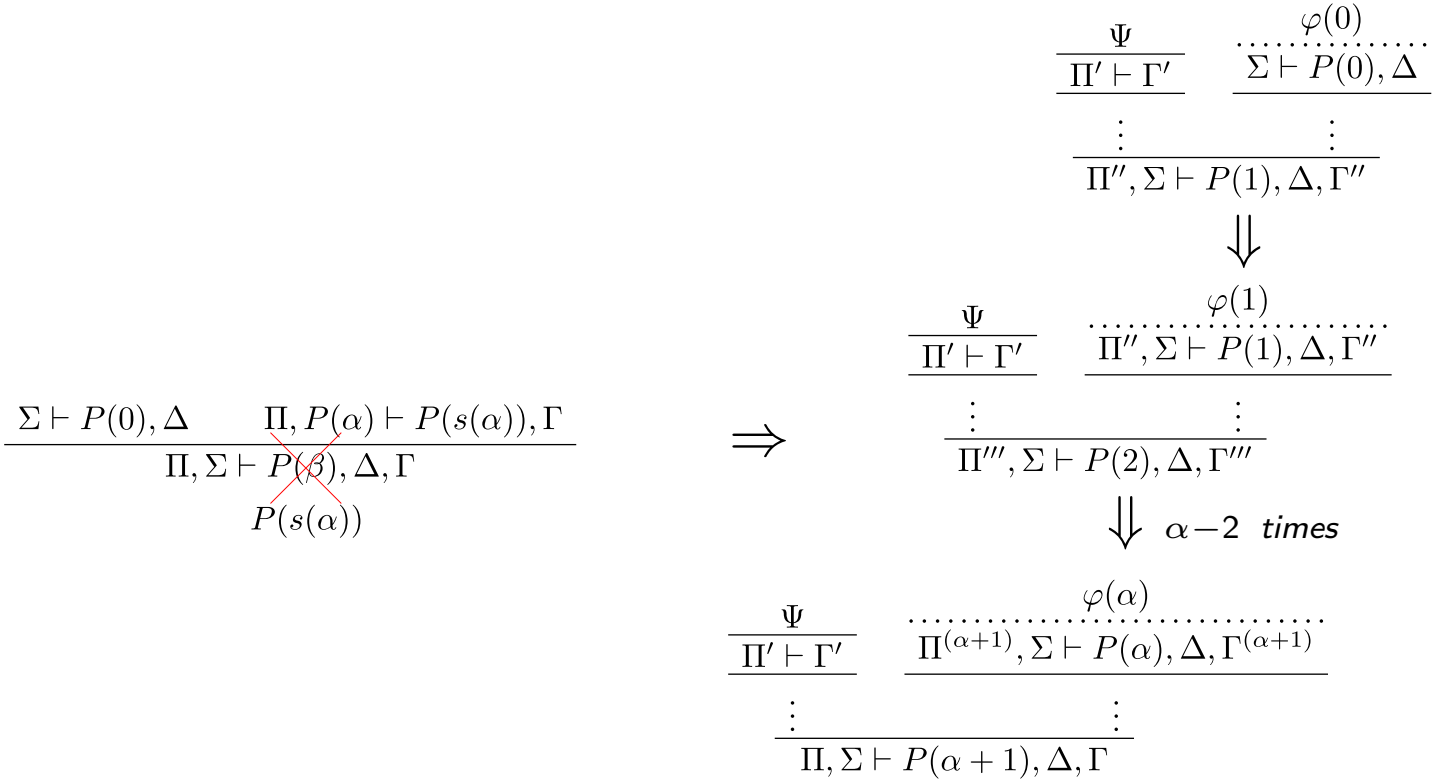
# Proof Schema as an Interpretation of Induction

- While initial work considered only a fragment of arithmetic, Schematic construction for Peano arithmetic are possible [ Cerna & Lolic , 2018].

$$\frac{\Sigma \vdash P(0), \Delta \qquad \Pi, P(\alpha) \vdash P(s(\alpha)), \Gamma}{\Pi, \Sigma \vdash P(\beta), \Delta, \Gamma} \quad P(s(\alpha)) \qquad \Rightarrow \qquad \dfrac{\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\varphi(0)}{\Sigma \vdash P(0), \Delta}}{\Pi'', \Sigma \vdash P(1), \Delta, \Gamma''} \Downarrow \dfrac{\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\varphi(1)}{\Pi'', \Sigma \vdash P(1), \Delta, \Gamma''}}{\Pi''', \Sigma \vdash P(2), \Delta, \Gamma'''}$$

# Proof Schema as an Interpretation of Induction

- While initial work considered only a fragment of arithmetic, Schematic construction for Peano arithmetic are possible [ Cerna & Lolic , 2018].



# Proof Schema as an Interpretation of Induction

- While initial work considered only a fragment of arithmetic, Schematic construction for Peano arithmetic are possible [ Cerna & Lolic , 2018].

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

$\Rightarrow$

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

$\Downarrow$

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

$\Downarrow$ $\alpha - 2$ *times*

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

- The proof is indexed by $\alpha$.
- Instantiating $\alpha$ results in an **LK**-proof .
- Formally a proof pair $\langle \varphi(0), \varphi(n+1) \rangle$.

# Sound Construction: The $\mathcal{S}i\mathbf{LK}$-Calculus

- The restrictions on links guaranteeing sound construction can be internalized by extending the calculus.
- Instead of sequents we consider sequent pairs ( $\top$ : **S** ) or ( **S**′ : [ **S** ] ) which can be open or closed.
- the left sequent of the component pair is dependent on the right sequent in the it defines its closure condition.
- Closure defines soundness, links alter the closure conditions through the following rules.

# $\mathcal{S}i\mathbf{LK}$-Calculus: Linking Rules

Table: The linking rules of the $\mathcal{S}i\mathbf{LK}$-calculus. For the other rules of the $\mathcal{S}i\mathbf{LK}$-Calculus see the Tableaux paper [Cerna & Lettmann 2017]

$$\frac{(\ \top : [\ (\Pi \vdash \Delta)[n \setminus 0]\ ]\ ), \Gamma | \dot{\Pi}}{(\ (\Pi \vdash^{(n+1)} \Delta)[\bar{x} \setminus \bar{t}] : [\ (\Pi \vdash \Delta)[n \setminus 0]\ ]\ ), \Gamma | \dot{\Pi}}\ \circlearrowleft$$

$$\frac{(\ \top : [\ \mathbf{S}\ ]\ ), \Gamma | \dot{\Delta} | (\ [\ (\Lambda \vdash \Gamma)[n \setminus h(n)]\ ] : [\ \mathbf{R}\ ]\ ) | \dot{\Pi}}{(\ (\Lambda \vdash^{f(n)} \Gamma)[n \setminus g(n)][\bar{y} \setminus \bar{t}] : [\ \mathbf{S}\ ]\ ), \Gamma | \dot{\Pi}'}\ \curvearrowright$$

Equational theory:

$$\mathcal{E} \equiv \{\widehat{f^0}(x) = x; \widehat{f^{s(n)}}(x) = f\widehat{f^n}(x)\}$$

Abbreviations:

$$\Delta \equiv P(0), \forall x.P(x) \to P(f(x)) \text{ and } \mathbf{S} \equiv \Delta \vdash P(\widehat{f^0(0)})$$

# $Si\mathbf{LK}$-Proof

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{(\ \top :\ P(0) \vdash P(0)\ )\mid}\,Ax_1 : r
}{\left(\ \top :\ P(0) \vdash P(\widehat{f^0}(0))\ \right)\mid}\,\mathcal{E}_1^{bc}
}{\left(\ \top :\ P(0), \forall x.P(x) \to P(f(x)) \vdash P(\widehat{f^0}(0))\ \right)\mid}\,(w : l)_1^{bc}
}{\left(\ \top :\ [\Delta \vdash P(\widehat{f^0}(0))\ ]\ \right)\mid}\,cl_{bc}
}{\left(\ P(f\widehat{f^n}(0)) \vdash^{s(n)} P(f\widehat{f^n}(0)) :\ [\ \mathbf{S}\ ]\ \right)\mid}\,Ax : l
}{(\ \top :\ [\ \mathbf{S}\ ]\ ), \left(\ P(f\widehat{f^n}(0)) \vdash^{s(n)} P(f\widehat{f^n}(0)) :\ [\ \mathbf{S}\ ]\ \right)\mid}\,br
}{\left(\ \Delta \vdash^{s(n)} P(\widehat{f^n}(0)) :\ [\ \mathbf{S}\ ]\ \right), \left(\ P(f\widehat{f^n}(0)) \vdash^{s(n)} P(f\widehat{f^n}(0)) :\ [\ \mathbf{S}\ ]\ \right)\mid}\,\circlearrowleft
}{\left(\ \Delta, P(\widehat{f^n}(0)) \to P(f\widehat{f^n}(0)) \vdash^{s(n)} P(f\widehat{f^n}(0)) :\ [\ \mathbf{S}\ ]\ \right)\mid}\,(\to : l)_2^{sc}
}{\left(\ \Delta, \forall x.P(x) \to P(f(x)) \vdash^{s(n)} P(\widehat{f^{s(n)}}(0)) :\ [\ \mathbf{S}\ ]\ \right)\mid}\,(\forall : l)_1^{sc}
}{\left(\ \Delta \vdash^{s(n)} P(\widehat{f^{s(n)}}(0)) :\ [\ \mathbf{S}\ ]\ \right)\mid}\,(c : l)_1^{sc}
}{\left(\ [\ \Delta \vdash P(\widehat{f^{s(n)}}(0))\ ] :\ [\ \mathbf{S}\ ]\ \right)\mid}\,cl_{sc}
$$

# Interpreting a Closed Set of Components

- A closed set of components, i.e. a proof schema can be interpreted as a fusion of multiple inductions [ Gentzen, 1969].
- However, Closed Set of Components have a so called leading component has evidenced by $\mathbf{Q}_0$ on the right.

$$\bigwedge_{i=0}^{m} \mathcal{I}(\mathbf{S}_i) \wedge \forall.x\Big( \bigwedge_{i=0}^{m} \big(\mathcal{I}(\mathbf{Q}_i\,[n \setminus x]) \rightarrow \mathcal{I}(\mathbf{Q}_i\,[n \setminus (x+1)])\big)\Big) \rightarrow \forall x.(\mathcal{I}(\mathbf{Q}_0\,[n \setminus x]),$$

- Extension to the proof schema for Peano arithmetic [ Cerna & Lolic , 2018] is currently being investigated.

# Proof Schema: by comparison

- Unlike formal systems using so called $\omega$-rules, primitive recursive construction is part of the object language.
- In contrast to cyclic proof formalisms, proofs are not by infinite descent, i.e. not regular infinite proof trees.
- Essentially proof schemata fall in between these two well known formalisms.
- The formalism allows easy tracking of formula occurrences.
- The ability to track formula occurrences provides interesting properties concerning cut-elimination.

# Global versus Local cut-elimination

**Local cut-elimination** reduces a cut formula's 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 versus Local cut-elimination

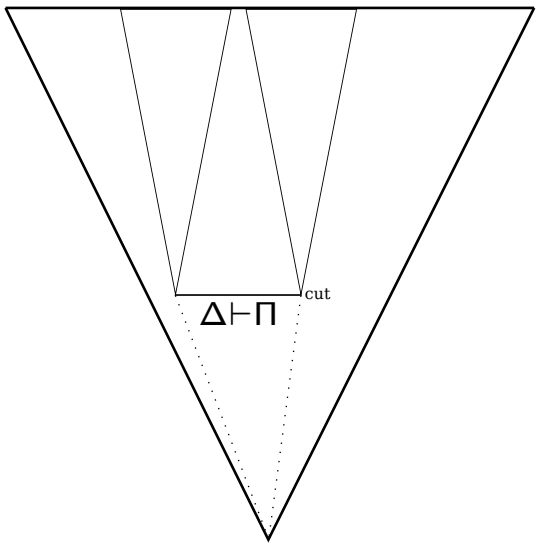
**Local cut-elimination** reduces a cut formula's complexity or its distance from the leaves.

- 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.

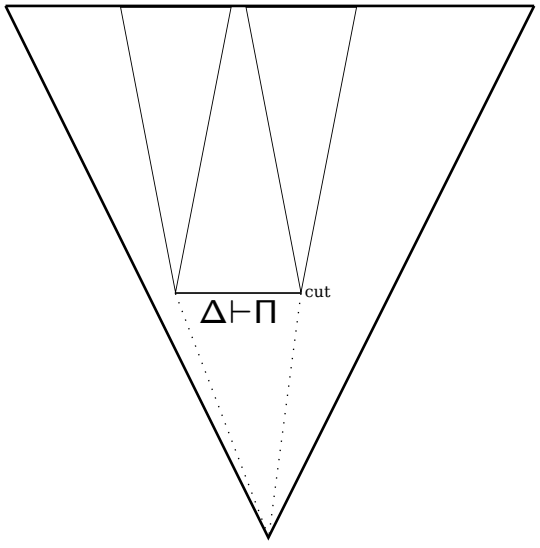
- From this intermediate representation a new proof with a **trivial cut-structure** is produced.

# CERES: The Characteristic Clause Set representation

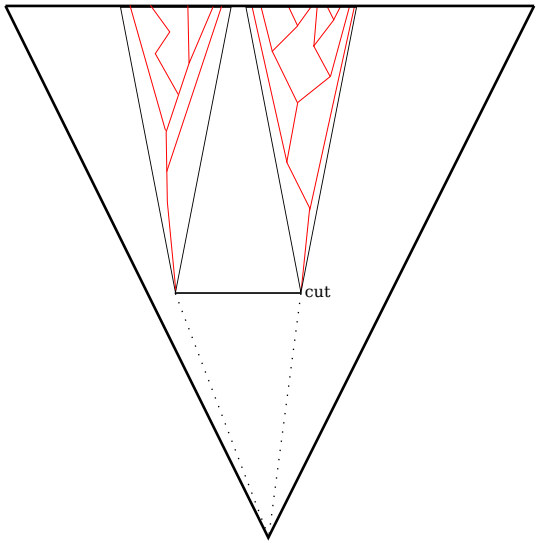



**LK**-Proof with cuts

# CERES: The Characteristic Clause Set representation

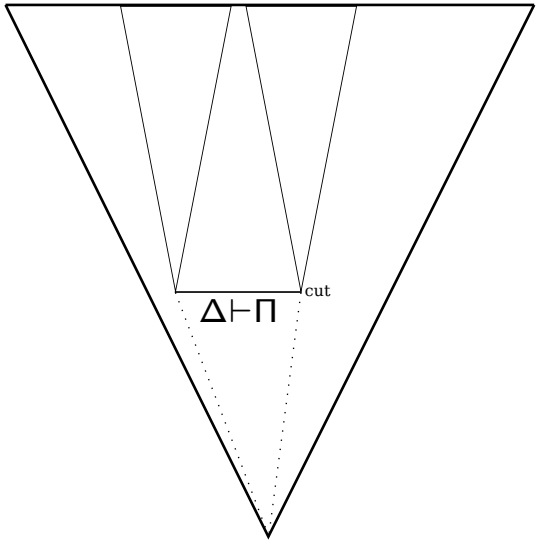



**LK**-Proof with cuts

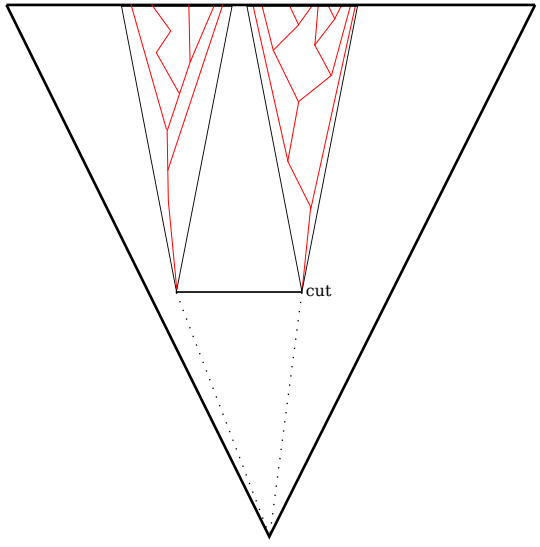



Paths to cut ancestors

# CERES: The Characteristic Clause Set representation




**LK**-Proof with cuts

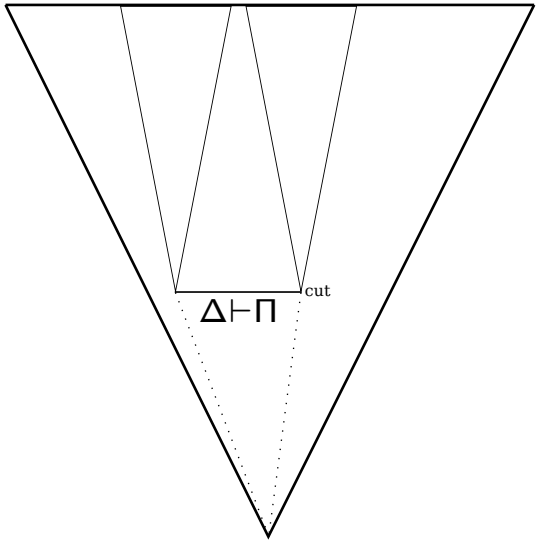



Paths to cut ancestors

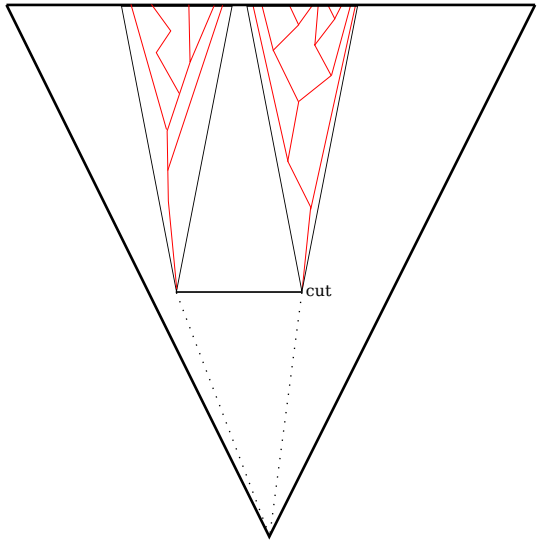
$$CL(A \vdash A) \equiv \{\vdash A\}$$
$$CL(A \vdash A) \equiv \{A \vdash\}$$
$$CL(A \vdash A) \equiv \{A \vdash A\}$$

- 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



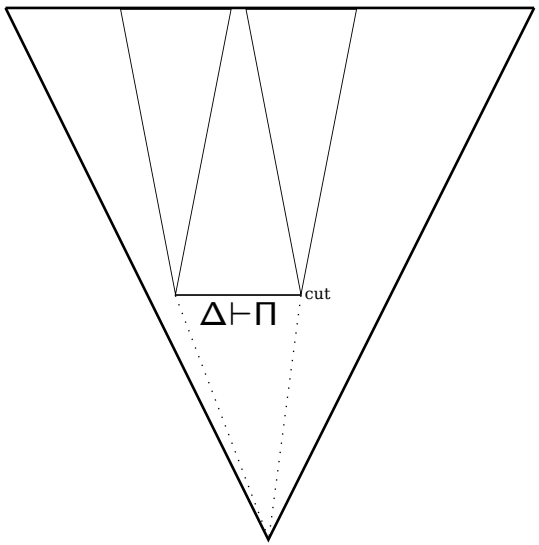

Paths to cut ancestors

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

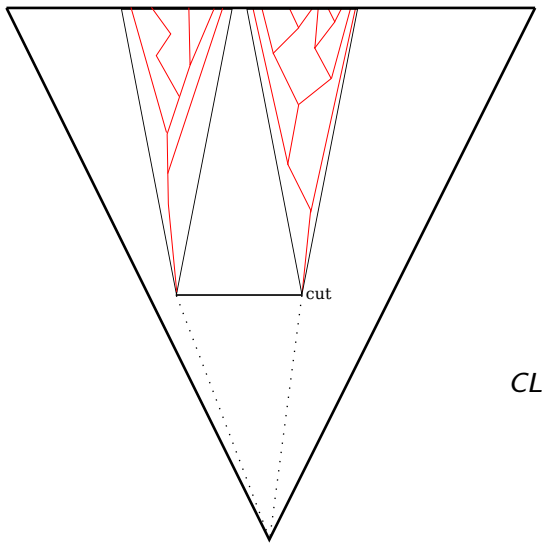
$$CL\left(\frac{\Delta \vdash \Pi}{\Delta' \vdash \Pi'}\ \rho\right) \equiv CL(\Delta \vdash \Pi)$$

- 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.

# Local Cut-elimination and Recursion

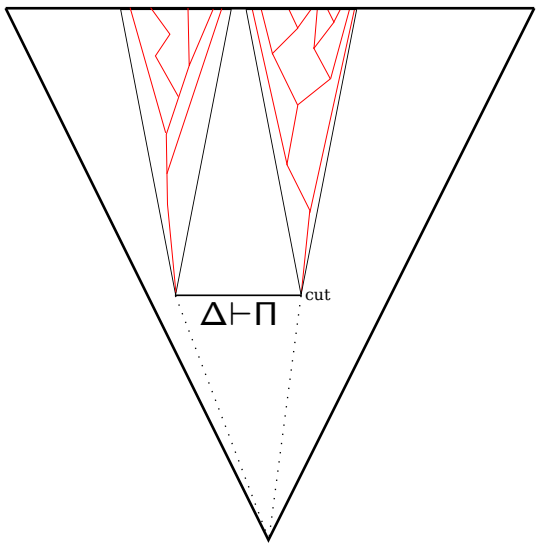
- Essentially cut reduction fails once it reaches a link (recursive call).

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

- Related formalisms eliminate cut from an infinite proof tree.
- One can define a relation between proof schema extending local cut-elimination and providing a sort of "cut-elimination" through clausal subsumption [Cerna & Lettmann 2017].

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

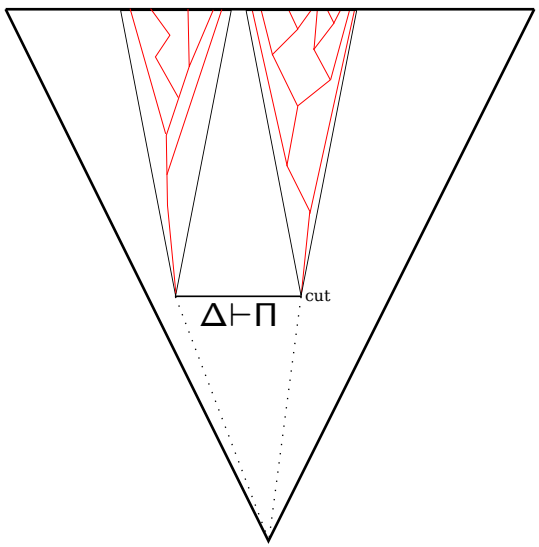
- Baaz and Leitsch, 2006 show how locally reducing cuts impacts the global cut structure.
- Every proof can be transformed into a proof with a minimally complex cut structure.
- The extracted clause set, is subsumed by the clause sets of the more complex cut structure.



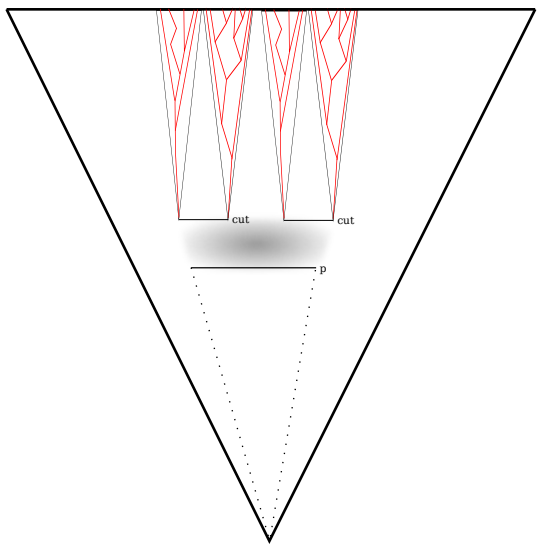

Reduction can result in

# 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.



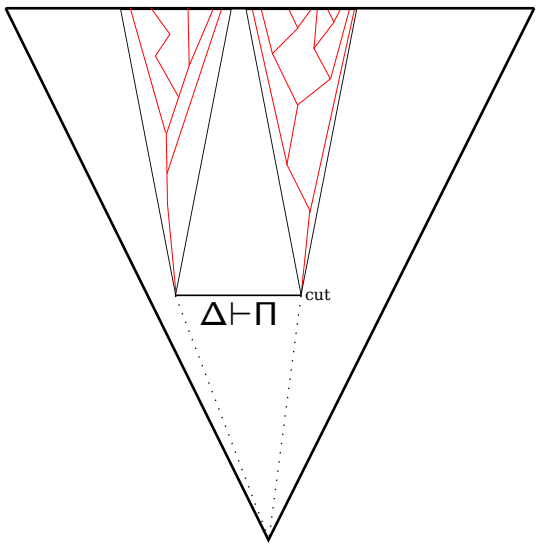

Reduction can result in



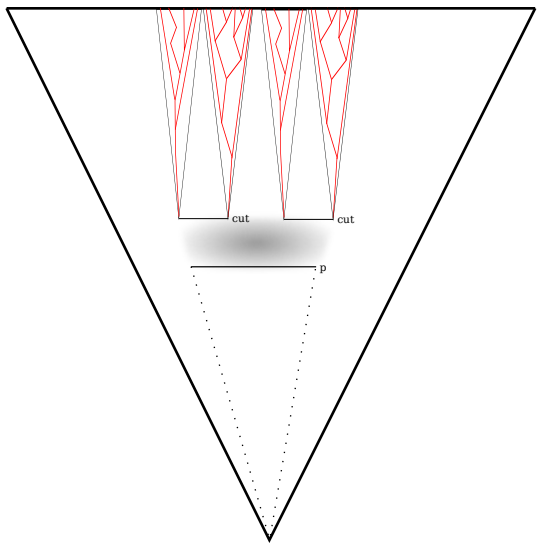

the following proof.

# 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

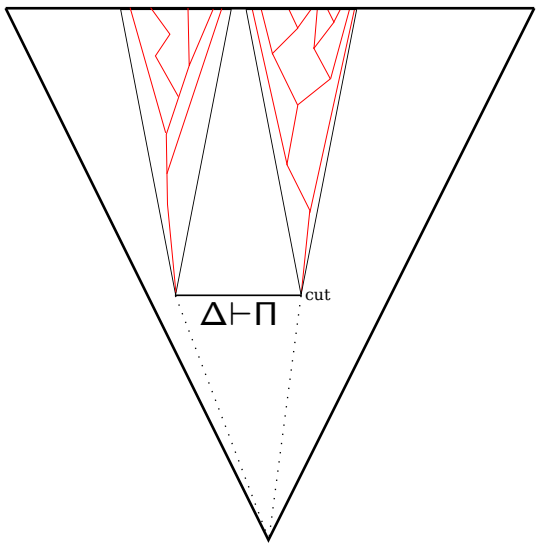



the following proof.

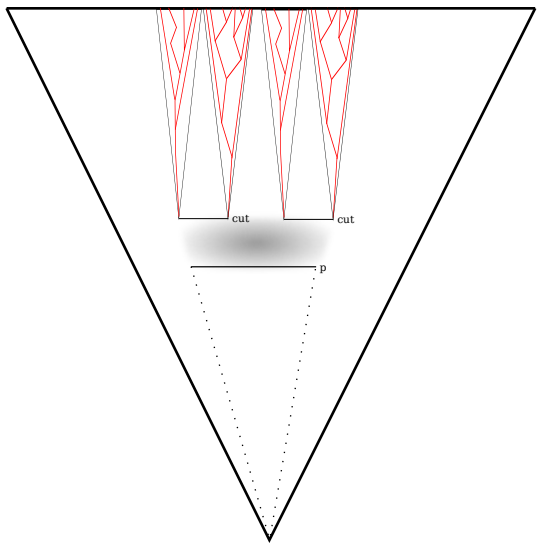
- Local elimination can result in a multiplication of the cuts
- Essentially, the cut-structure gets more redundant.

# 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.



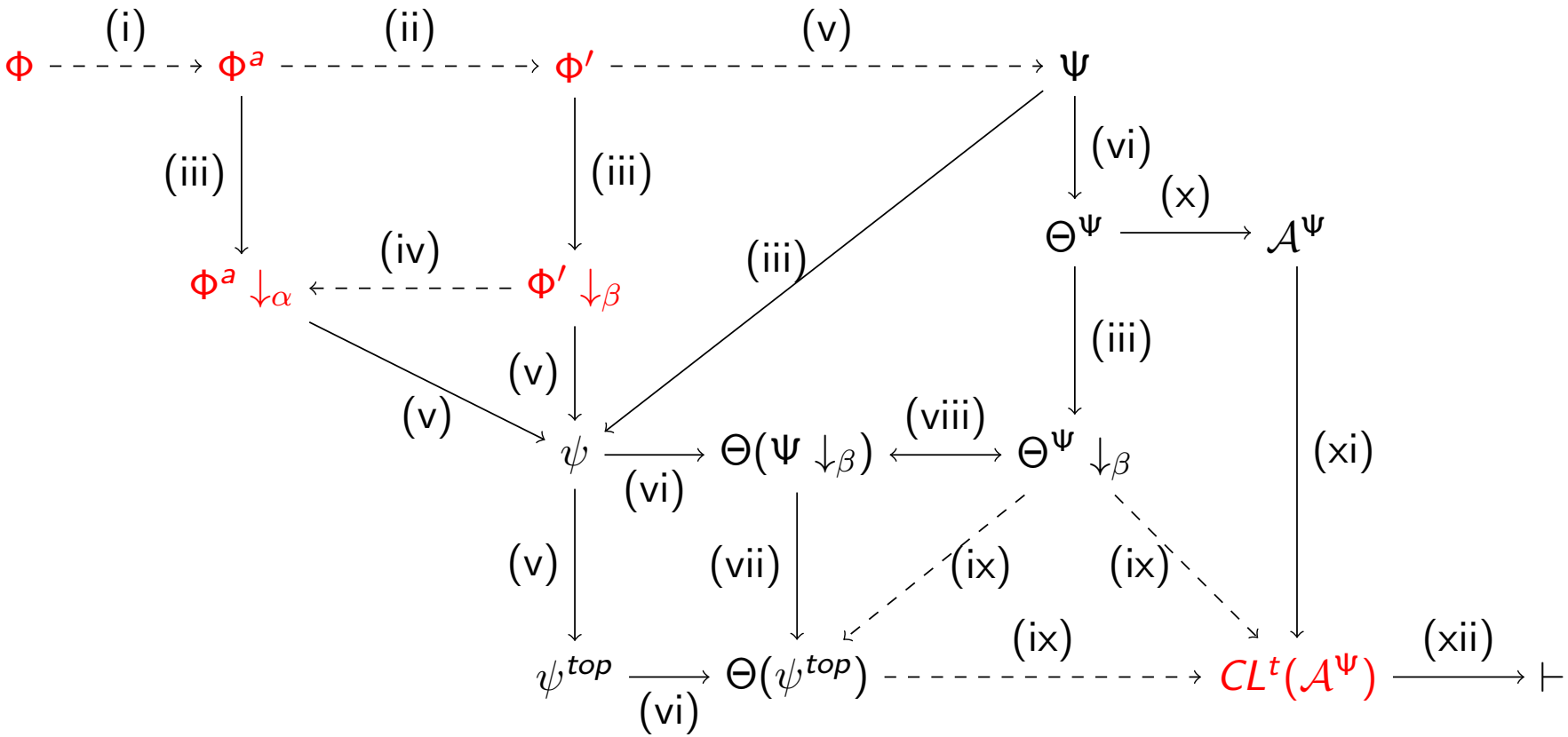

Reduction can result in




the following proof.

- Local elimination can result in a multiplication of the cuts
- Essentially, the cut-structure gets more redundant.
- Redundancy ≡ **structural simplicity**.

# Clausal Analysis of Proof Schema



# Global Cut-elimination and Proof Schema

- Recursive clausal analysis provides insight into the structure of proof schema.
- Though it uses infinite constructions similar to other formalisms
- Also, a recursive description of formula occurrences is loss.
- In [ Leitsch et al., 2017] a solution is provided which preserves the occurrence tracking properties.

# A Normal Form

$$\dfrac{\dfrac{\phi_\alpha}{\Gamma \vdash \Delta, F_\alpha} \quad \vdots}{\Gamma \vdash \Delta}(cut + c^*)$$

$$\dfrac{\dfrac{\phi_2}{\Gamma \vdash \Delta, F_2} \quad \dfrac{\dfrac{\phi_1}{\Gamma \vdash \Delta, F_1} \quad \dfrac{\dfrac{\Phi}{F_1, \ldots, F_\alpha \vdash}(w : l)}{F_1, \ldots, F_\alpha, \Gamma \vdash \Delta}(cut + c^*)}{\Gamma, F_2, \ldots, F_\alpha \vdash \Delta}(cut + c^*)}{\Gamma, F_3, \ldots, F_\alpha \vdash \Delta}$$

- The cut structure is turned into a recursively defined formula based on subformula occurance (**BLUE**).
- The schema itself is transformed into a schema with the cut structure as a formula in the consequent (**RED**).
- The formula is $\Sigma_1$ and unsatisfiable. The sequence $F_1, \ldots, F_\alpha$ contain the term tuples of a Schematic Herbrand Sequent.

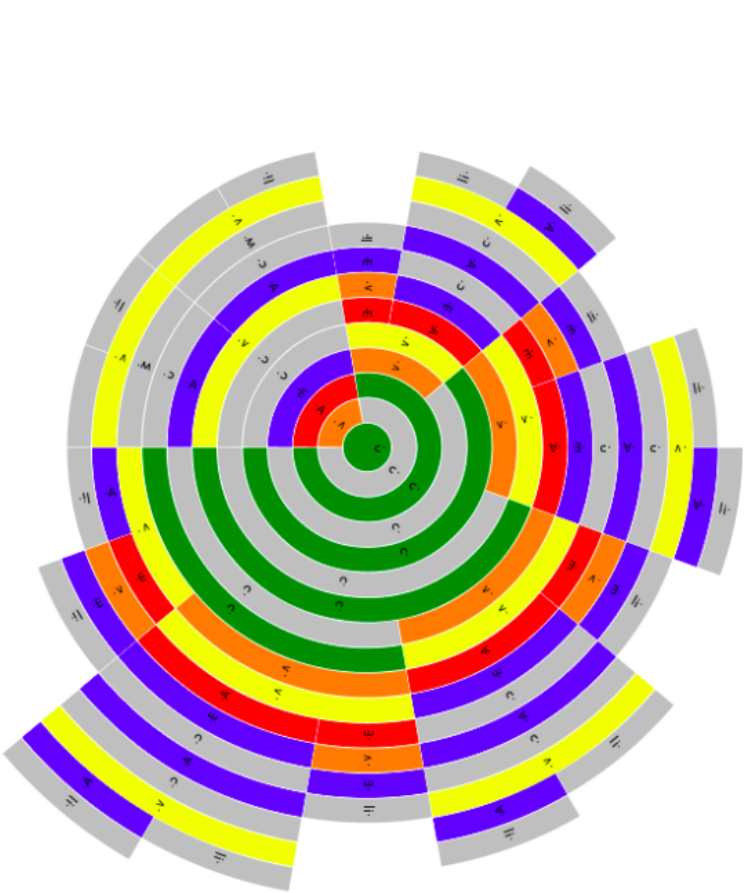
# Example Cut Structure definition: 1-SMA

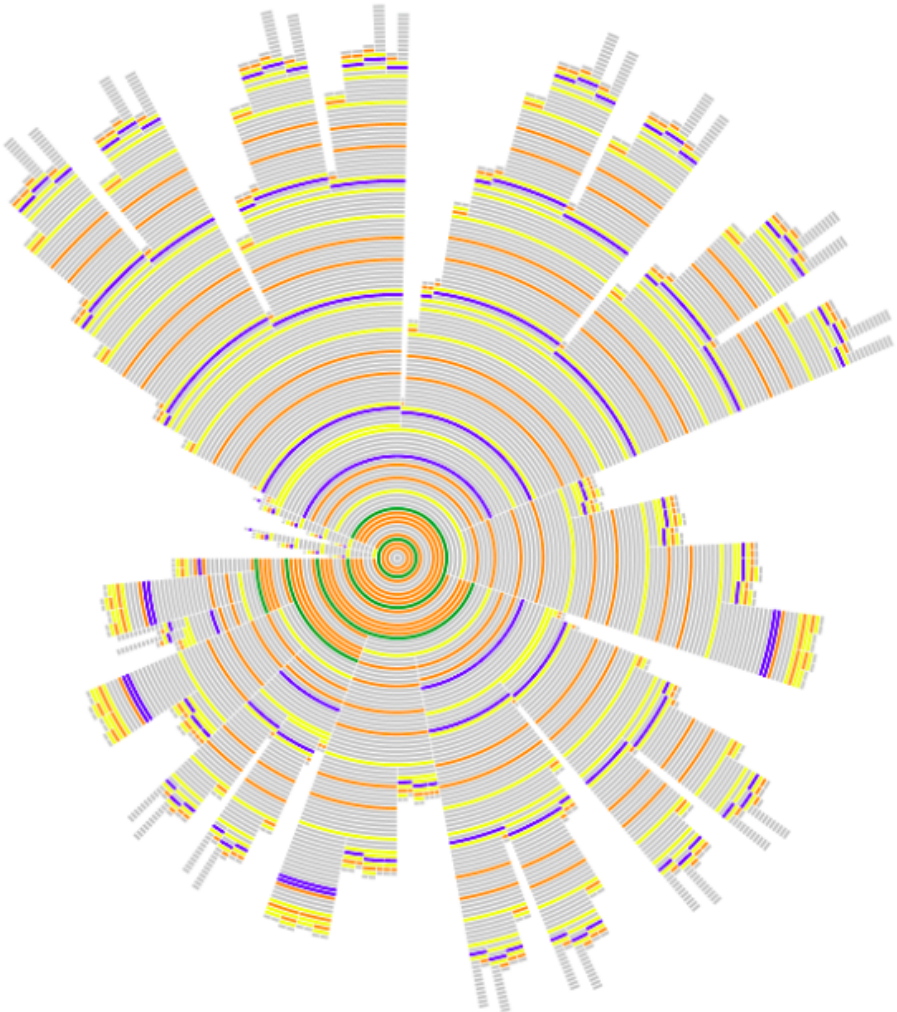
$$
\begin{aligned}
\mathit{Top}(0) &= \mathit{Next}(0) \wedge (0 = f(\mathbf{0}) \vee 0 = f(\mathbf{S}(\mathbf{0}))) \\
\mathit{Top}(n+1) &= \forall x((n+1) = f(\mathbf{S}(x)) \vee f(x) < (n+1)) \wedge \\
&\quad \forall x((n+1) = f(x) \vee f(x) < (n+1)) \wedge \mathit{Next}(n+1) \\
\mathit{Next}(0) &= (\neg f(\mathbf{0}) < 0) \wedge \forall x((\neg 0 = f(x)) \vee (\neg 0 = f(\mathbf{S}(x)))) \\
\mathit{Next}(n+1) &= \forall x((\neg(n+1) = f(x)) \vee (\neg(n+1) = f(\mathbf{S}(x)))) \wedge \\
&\quad \forall x((\neg f(x) < (n+1)) \vee n = f(x) \vee f(x) < n) \wedge \\
&\quad \forall x((\neg f(\mathbf{S}(x)) < (n+1)) \vee n = f(\mathbf{S}(x)) \vee f(x) < n) \\
&\quad \wedge \mathit{Next}(n)
\end{aligned}
$$

# 1-SMA: Important Properties

- Unlike the previous examples $\forall n Top(n) \vdash$ The proof structure is well understood and the minimal Herbrand sequent is know! [ D. M. Cerna, 2018, under review]
- Viper can prove this statement in roughly 5 hours
- The superposition prover [ Aravantinos *et al.* , 2013] Can prove 1-SMA in roughly 0.01 seconds but cannot prove k-SMA on theoretic grounds.
- Examples like this may aid our understanding of inductive theorem proving.

# 1-SMA: Proof & Refutation





# Conclusion & Future Work

- Proof schema provide an alternative formalism for reasoning by induction.
  - Generalizations of existing techniques can easily be defined.
  - Transformation and Analysis of proofs results in interesting and complex theorem proving problems.
  - Around 60 problems have been added to the TPTP library and currently working on a submission to TIP.
- Investigation into the relationship between proof Schema and cyclic proofs is currently of interesting.
  - Does each proof schema have a cyclic proof as it's limit?
  - Can schematic proof transformation benefit from cyclic proofs?
- Are there theoretic problems which can benefit from the formalism?
  - For example, [ Cerna & Lolic , 2018] provide a non-trivial conservative reflection between the schematic and **LK**calculus for Peano arithmetic.

Thank you for your time.