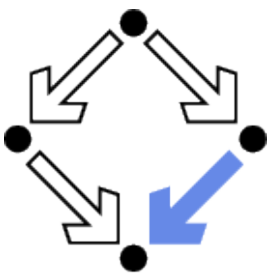
Towards the Automatic Construction of Schematic Proofs: A Motivation for the Schematic Approach.

David M. Cerna and Michael Lettmann

July 7th, 2018

JOHANNES KEPLER
UNIVERSITY LINZ



Proof Schema: a.k.a Yet Another Formalism of Induction

- A schema of proofs was used to analyze Fürstenberg's proof of the infinitude of primes [Baaz et al. 2008].
- Proof Schema are a formal description of this representation.
- More precisely, they are recursively defined infinite sequence of finite proofs indexed by a vector of free numeric parameters, which when grounded and normalized produce a first-order proof.
- Links between proofs define the recursive construction.
- Recent work [Cerna & Lolic , 2018], has shown equivalence to Peano Arithmetic.

Proof Schema: by comparison

- Unlike formal systems using so called ω-rules, primitive recursive construction is an explicit part of the object language.
- In contrast to cyclic proof formalisms, proofs are not by infinite descent, i.e. they do not unroll into 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

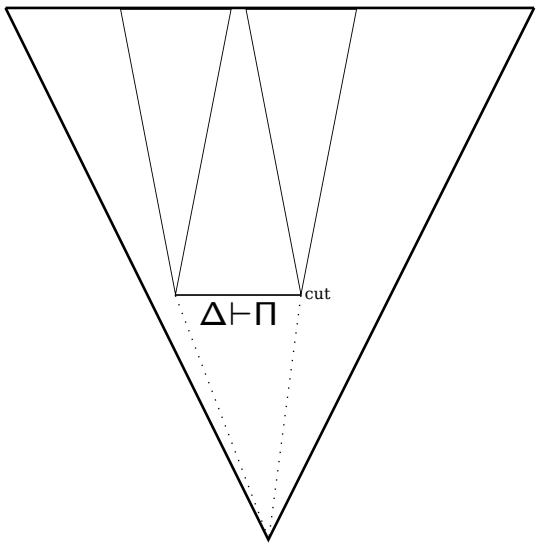
Local cut-elimination reduces a cut formula's complexity or its distance from the leaves.

- 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.

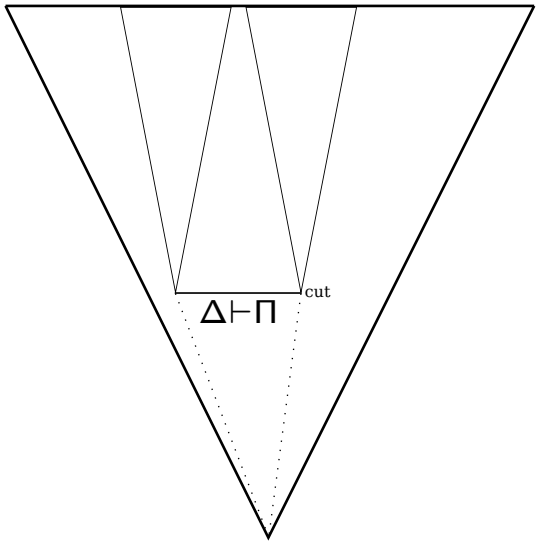
- From this intermediate representation a new proof with a **trivial cut-structure** is produced.

CERES: The Characteristic Clause Set representation

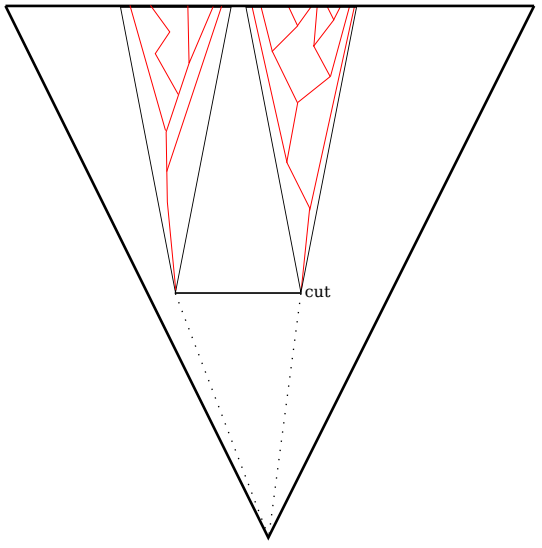


LK-Proof with cuts

CERES: The Characteristic Clause Set representation

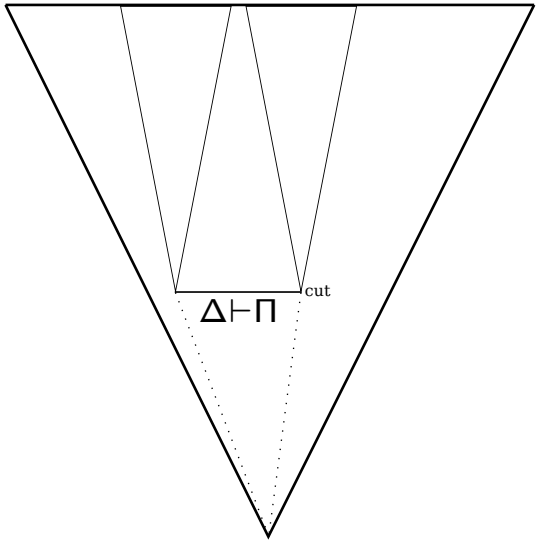


LK-Proof with cuts

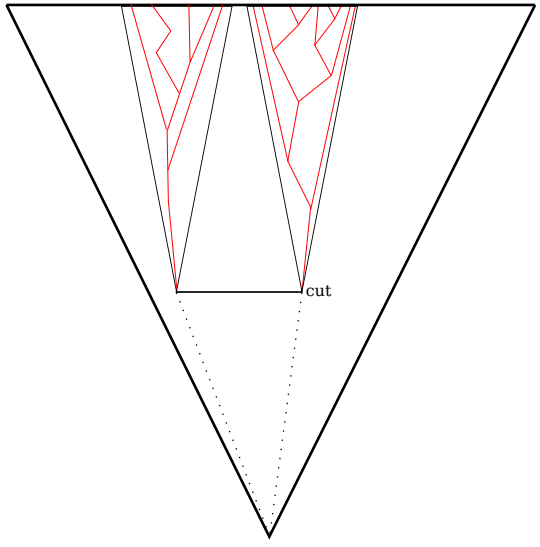


Paths to cut ancestors

CERES: The Characteristic Clause Set representation



LK-Proof with cuts

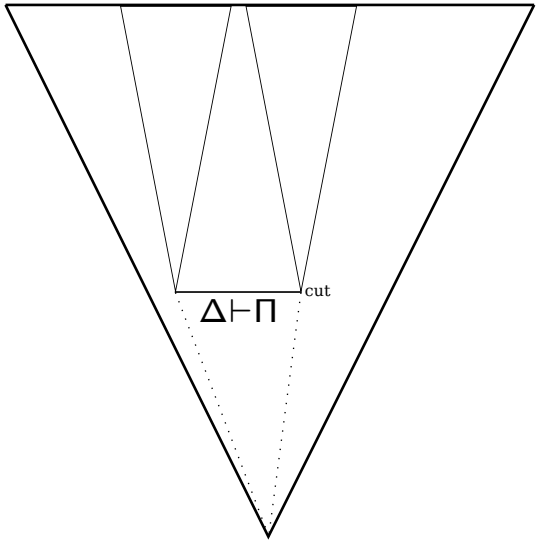


Paths to cut ancestors

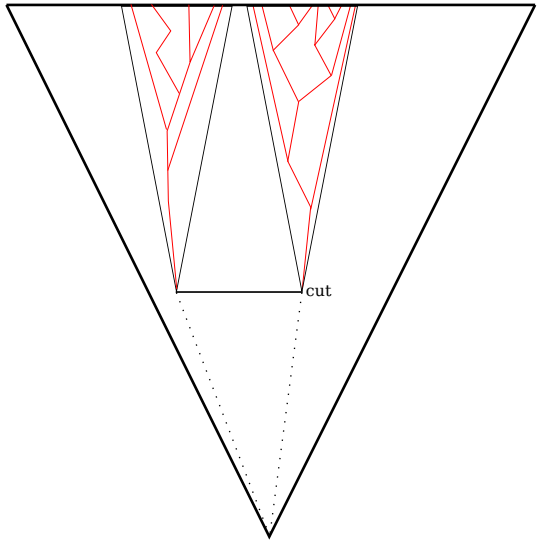
$$CL(A \vdash A) \equiv \{\vdash A\}$$
$$CL(A \vdash A) \equiv \{A \vdash\}$$
$$CL(A \vdash A) \equiv \{A \vdash A\}$$

- 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



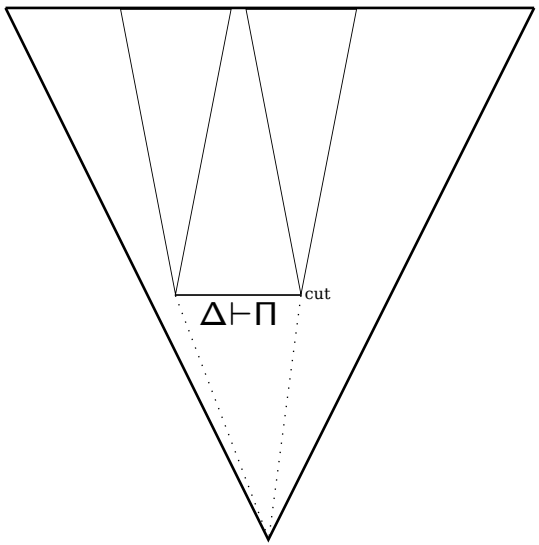
Paths to cut ancestors

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

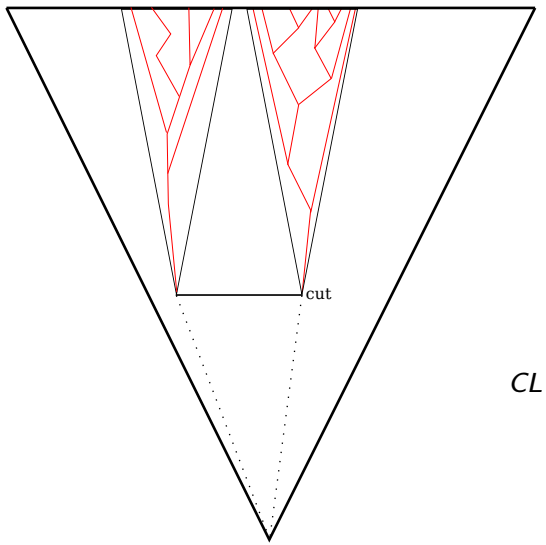
$$CL\left(\frac{\Delta \vdash \Pi}{\Delta' \vdash \Pi'}\ \rho\right) \equiv CL(\Delta \vdash \Pi)$$

- 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

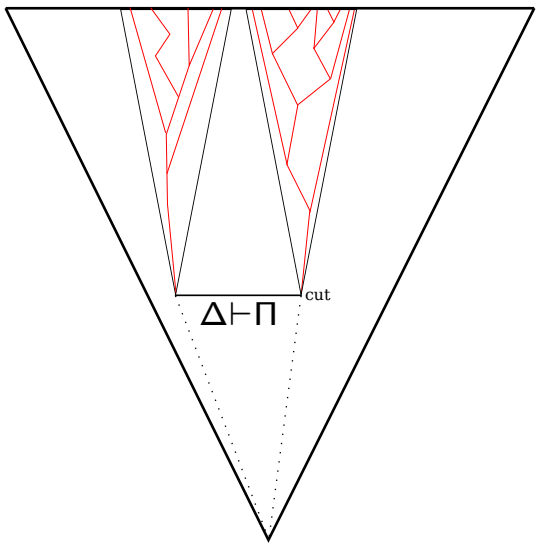
- Essentially cut reduction fails once it reaches a link (recursive call).

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

- For related formalisms cuts are eliminated 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

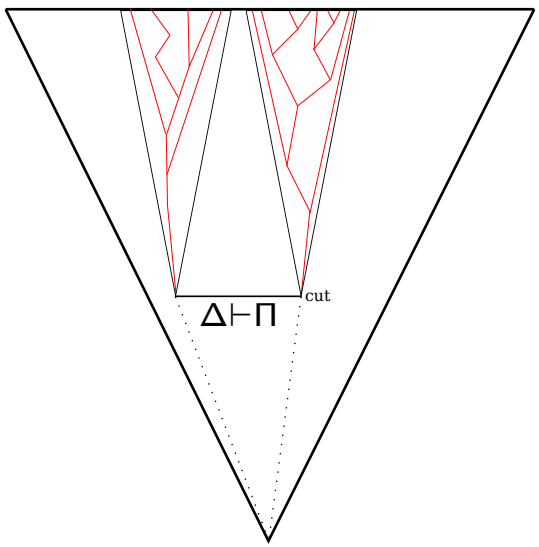
- Baaz and Leitsch, 2006 show how locally reducing cuts impacts the global cut structure.
- Every proof can be transformed into a proof with a minimally complex cut structure.
- The extracted clause set, is subsumed by the clause sets of the more complex cut structure.



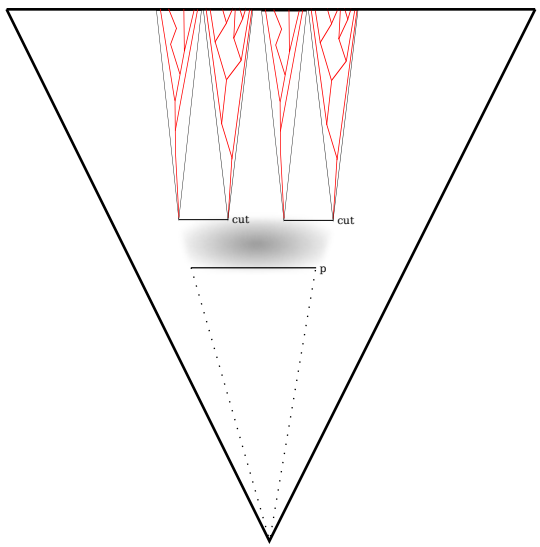
Reduction can result in

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.



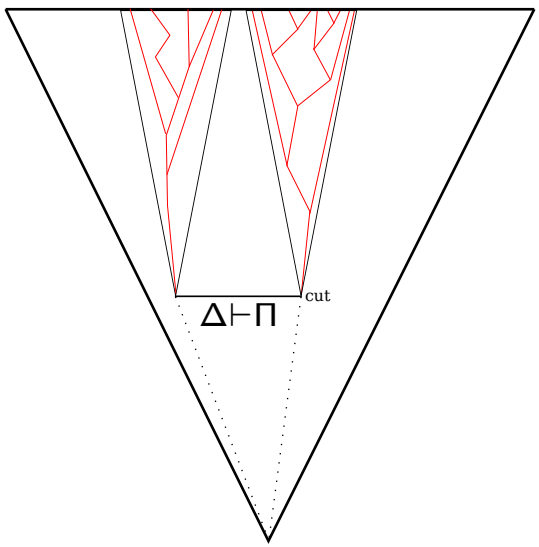
Reduction can result in



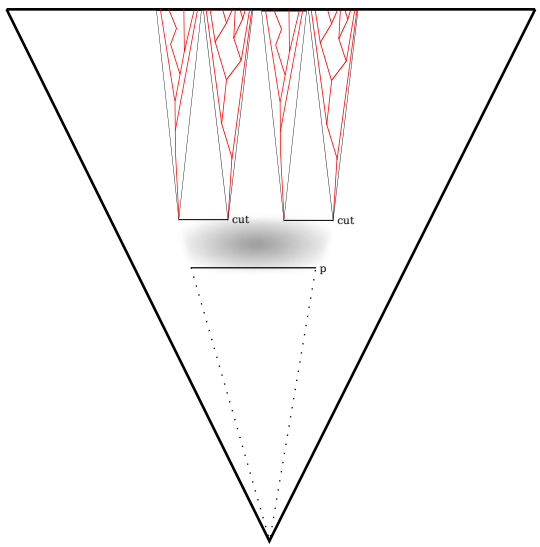
the following proof.

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

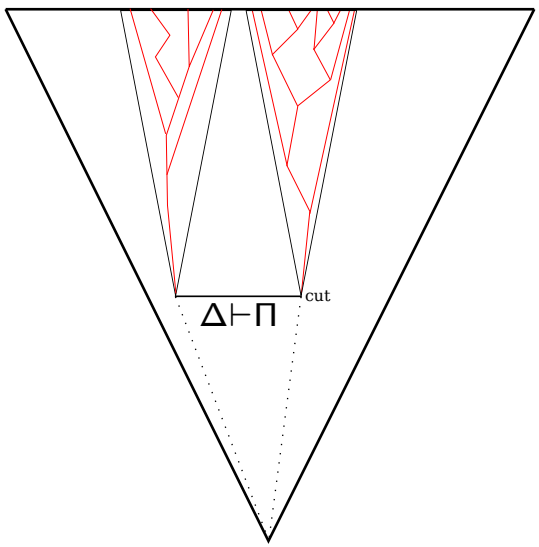


the following proof.

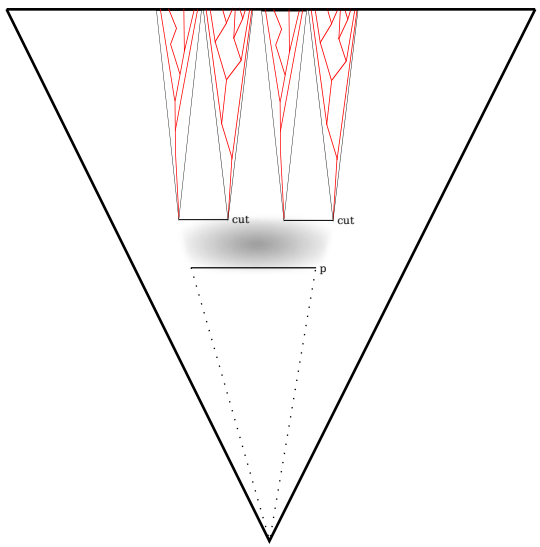
- Local elimination can result in a multiplication of the cuts
- Essentially, the cut-structure gets more redundant.

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.



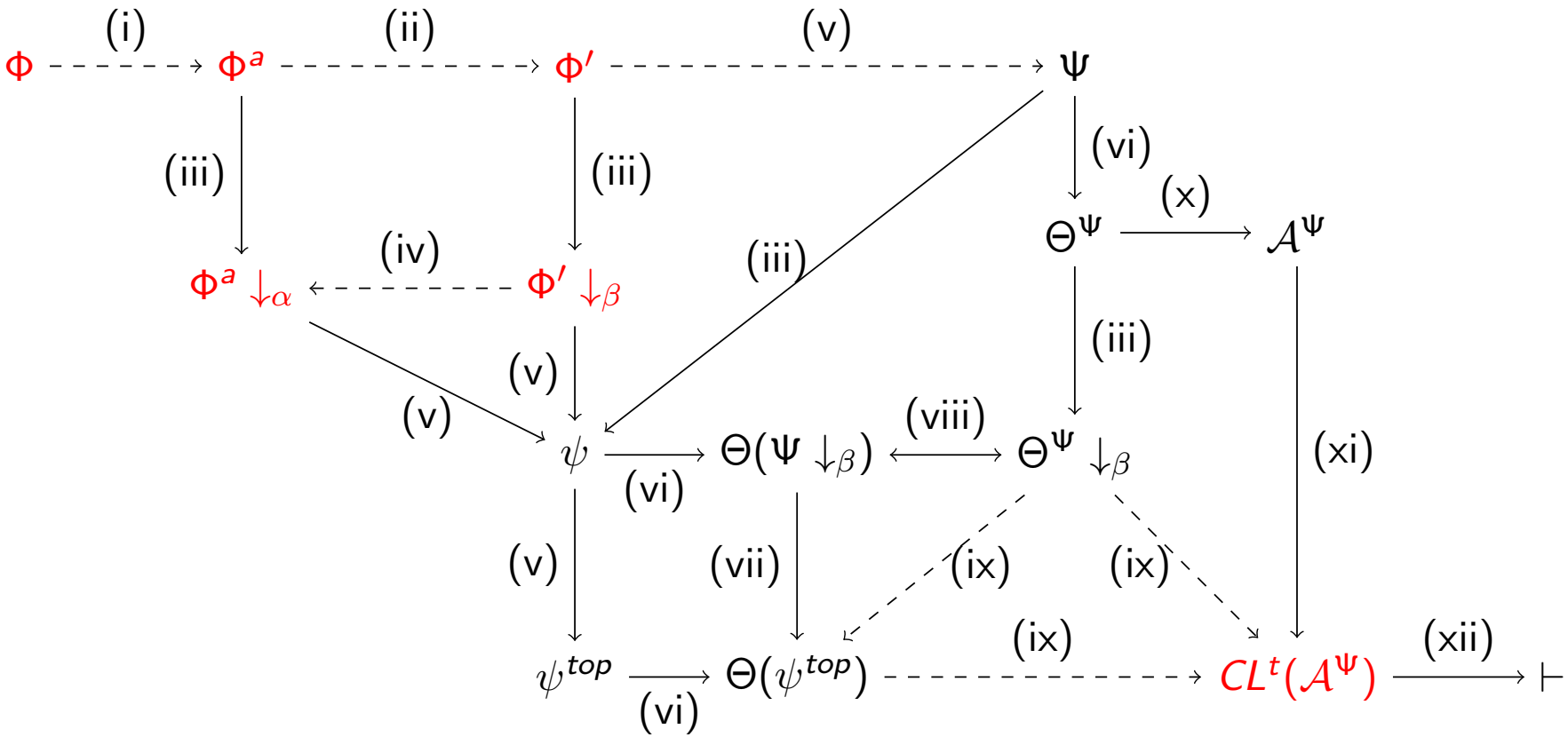
Reduction can result in



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 some insight into the structure of proof schema.
- Though it is too close for comfort to the infinite constructions of 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 Σ_1 and unsatisfiable. The sequence $F_1, \ldots, F_\alpha$ contain the term tuples of a Schematic Herbrand Sequent.

Cyclic Proofs?

- F is an inductive definition of an unsatisfiable Σ_1 formula.
- Current tools we us to find proofs of $\forall \bar{x} F(n_1, \cdots, n_m, x) \vdash$ are not sufficient for our more complex examples.
- Currently investigating how cyclic proving methods can help.
- For example Realizability in Cyclic Proofs [R. Rowe and J Brotherston, 2017]

Example definition: IPP_ω

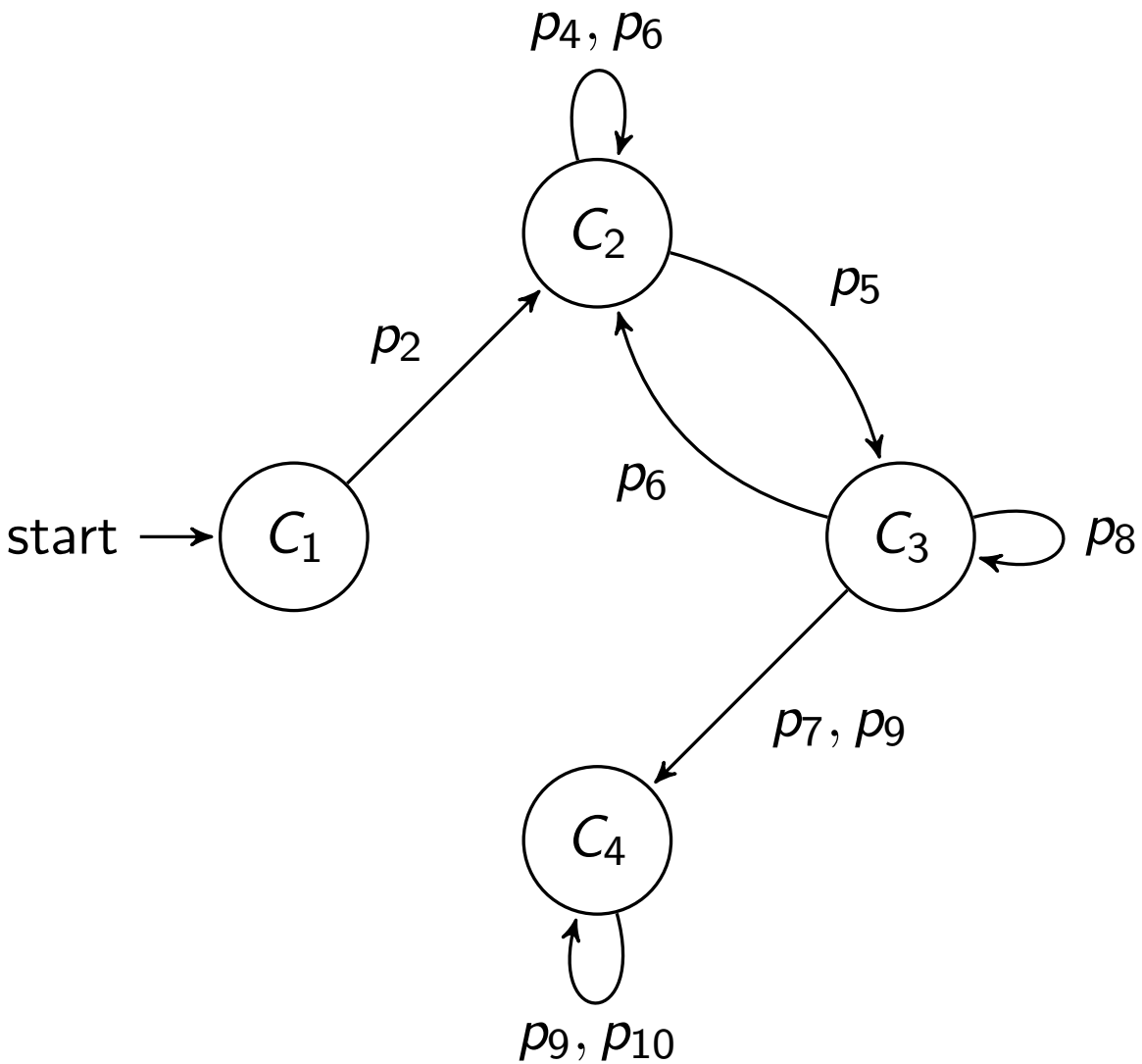
$$
\begin{aligned}
R(s(n)) \Longrightarrow\ & \forall x(\ x \leq x\ )\ \wedge\ T(s(n))\ \wedge\ \forall x(\ Q(s(n), x)\ ) \\
R(0) \Longrightarrow\ & \forall x, y(\neg(s(y) \leq x\ \wedge\ f(y) = 0\ \wedge \\
& f(x) = 0))\ \wedge\ \forall x(\ x \leq x\ ) \wedge \\
& \forall x(f(x) = 0)
\end{aligned}
$$

$$
\begin{aligned}
T(s(n)) \Longrightarrow\ & \forall x, y, z(\ m(x, y) \leq z\ \Longrightarrow\ x \leq z\ ) \wedge \\
& \forall x, y, z(\ m(x, y) \leq z\ \Longrightarrow\ y \leq z\ ) \wedge \\
& \forall x, y(\neg(s(y) \leq x\ \wedge\ f(y) = s(n) \wedge f(x) = s(n))) \wedge \\
& T(n) \\
T(0) \Longrightarrow\ & \forall x, y(\neg(s(y) \leq x\ \wedge\ f(y) = 0\ \wedge\ f(x) = 0)) \\
Q(s(n), a) \Longrightarrow\ & f(a) = s(n) \vee Q(n, a) \\
Q(0, a) \Longrightarrow\ & f(a) = 0
\end{aligned}
$$

Refuting the IPP_ω Definition

- Constructing a formal proof of $\forall x R(n, x) \vdash$ is quite difficult.
- Formalizing proofs and producing the term tuples of a schematic Herbrand sequent of the cut structure definitions are two areas of current investigation.
- In recent work [D. M. Cerna, 2018], currently in review, a formal system, which seems to be complete (still an open question), was introduced which can formalize $\forall x R(n, x) \vdash$.
- The following diagram represents the linking dependencies of the formal proof.

Refutation Link Dependences



Constructing Schematic Proofs

- We have mentioned proof schema but have not really mentioned their construction:

Constructing Schematic Proofs

- We have mentioned proof schema but have not really mentioned their construction:

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

Constructing Schematic Proofs

- We have mentioned proof schema but have not really mentioned their construction:

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

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

Constructing Schematic Proofs

- We have mentioned proof schema but have not really mentioned their construction:

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

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

Constructing Schematic Proofs

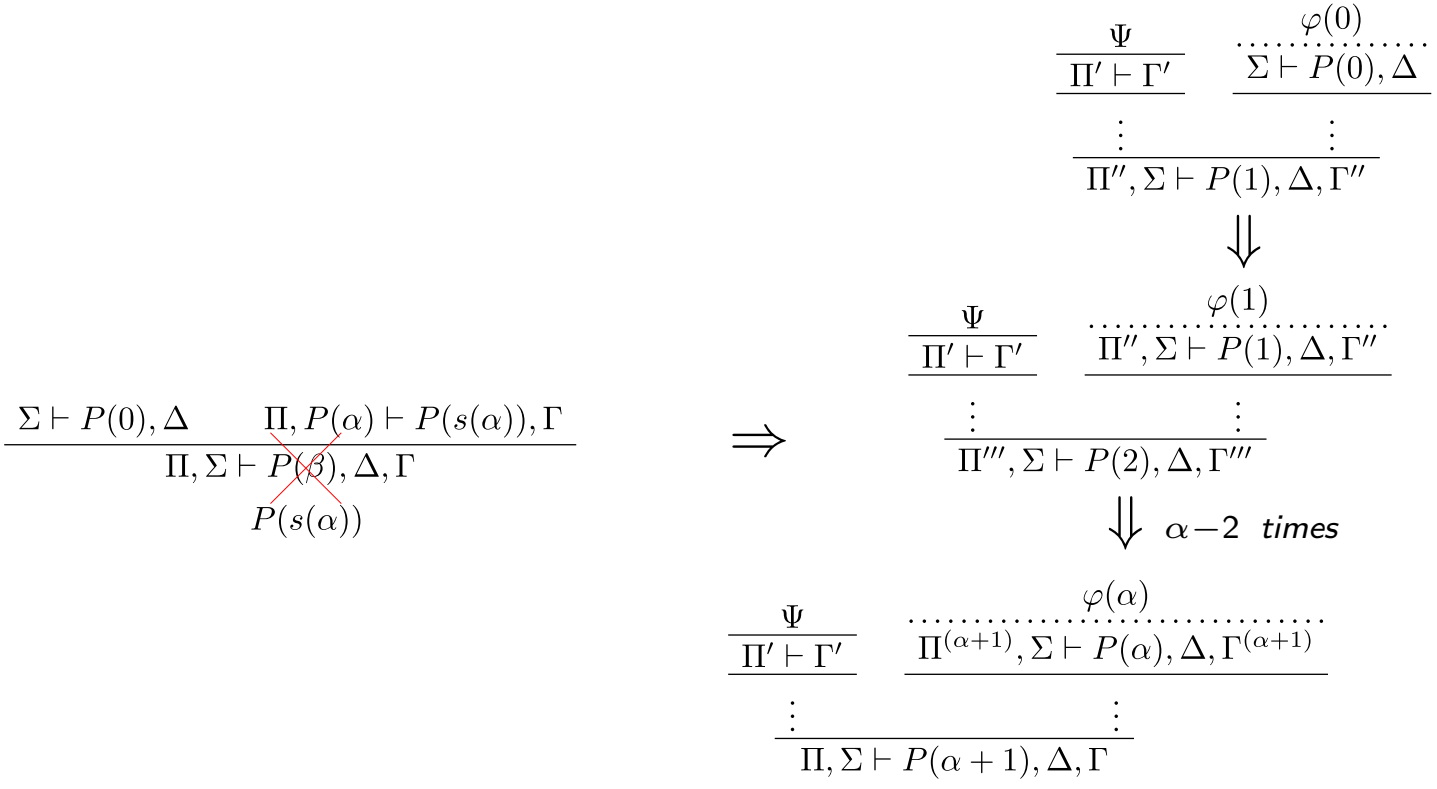
- We have mentioned proof schema but have not really mentioned their construction:

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

$$\dfrac{\dfrac{\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\varphi(0)}{\Sigma \vdash P(0), \Delta}}{\Pi'', \Sigma \vdash P(1), \Delta, \Gamma''}}{\Pi''', \Sigma \vdash P(2), \Delta, \Gamma'''} \qquad \Downarrow \qquad \dfrac{\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\varphi(1)}{\Pi'', \Sigma \vdash P(1), \Delta, \Gamma''}}{\Pi''', \Sigma \vdash P(2), \Delta, \Gamma'''}$$

Constructing Schematic Proofs

- We have mentioned proof schema but have not really mentioned their construction:



Constructing Schematic Proofs

- We have mentioned proof schema but have not really mentioned their construction:

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

$\Rightarrow$

$$\dfrac{\dfrac{\dfrac{\Psi}{\Pi' \vdash \Gamma'} \qquad \dfrac{\varphi(0)}{\Sigma \vdash P(0), \Delta}}{\Pi'', \Sigma \vdash P(1), \Delta, \Gamma''} \Downarrow}{\Pi''', \Sigma \vdash P(2), \Delta, \Gamma'''} \Downarrow \alpha - 2 \text{ times}$$

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

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

$\mathcal{S}i\mathbf{LK}$-Calculus: Objects

- Sequents: $\Pi \vdash \Sigma$ or $\mathbf{S}$
- Component pairs: $(\top : \mathbf{S})$ or $(\mathbf{S}' : [\mathbf{S}])$
  Closed component pairs contain the end-sequents of a proof schema component
- Component groups: multisets of component pairs Γ
- Component collections: sets of component groups $\dot{\Pi}$

$\mathcal{S}i\mathbf{LK}$-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

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

Evaluation Function of a Closed Component Collection

- Let $\mathcal{I}$ be the customary evaluation function of sequents.
- Assume an $\mathcal{S}i\mathbf{LK}$-proof ending in the component collection

$$\mathbf{C} \equiv (\ [\ \mathbf{Q}_0\ ]\ :\ [\ \mathbf{S}_0\ ]\ ) \,|\, \cdots \,|\, (\ [\ \mathbf{Q}_m\ ]\ :\ [\ \mathbf{S}_m\ ]\ )$$

such that $(\ [\ \mathbf{Q}_0\ ]\ :\ [\ \mathbf{S}_0]\ )$ is the last component pair closed in the proof of C (leading component).

$$\mathcal{I}_{\mathcal{S}i\mathbf{LK}}(\mathbf{C}) \equiv \mathcal{I}(\mathbf{S}_0),$$

if $\mathbf{Q}_0 \equiv [\ ]$ and $\mathcal{I}_{\mathcal{S}i\mathbf{LK}}(\mathbf{C}) \equiv$

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

A Better $\mathcal{S}i$**LK**-Calculus

- $\mathcal{S}i$**LK**-Calculus was not developed for the formalism presented in [Cerna & Lolic , 2018] (**PA** Schema).
- However, there is a no reason it cannot be extended to so called P-schema (**PA** Schema).
- This can be done by introducing Super sequents or sequents of sequents.
- The super antecedent represents the assumed theory and the super consequent represents the statements implied by the antecedent. An empty antecedent means it is a definitional statement Classical logic plus our equational theory.
- The rules of the $\mathcal{S}i$**LK**-Calculus can be generalized and simplified in this setting.

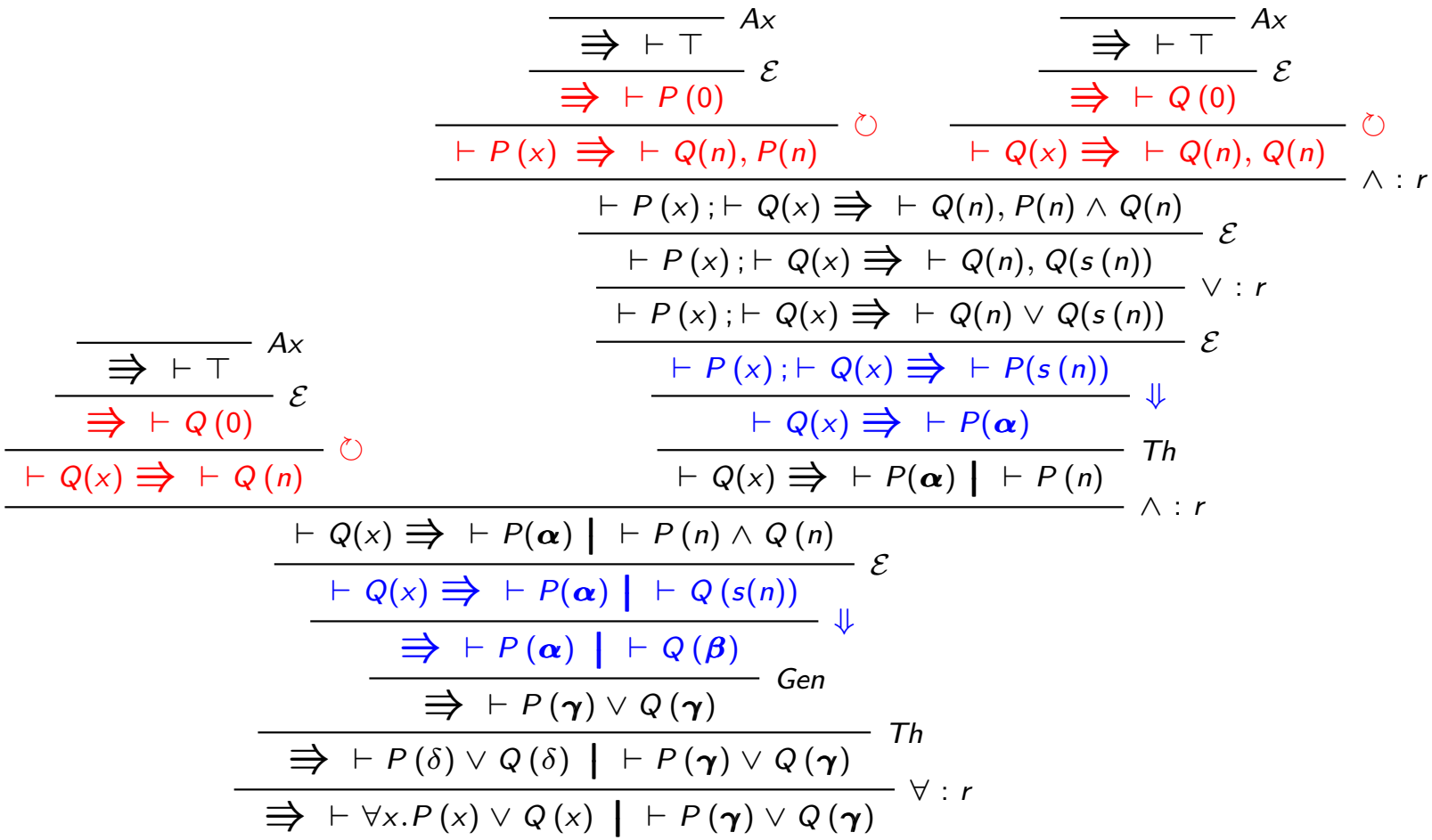
The Super Sequent Calculus

$$\frac{}{\Rrightarrow \Gamma, A \vdash A, \Delta}\, Ax \qquad \frac{}{\Rrightarrow \Gamma \vdash \top, \Delta}\, Ax$$

$$\frac{\mathscr{F} \Rrightarrow \mathscr{G} \mid S(0)}{S(x) \mid \mathscr{F} \Rrightarrow \mathscr{G} \mid S(m) \circ T^*}\, \circlearrowleft \qquad \frac{S(x) \mid \mathscr{F} \Rrightarrow \mathscr{G} \mid S(s(m))}{\mathscr{F} \Rrightarrow \mathscr{G} \mid S(\alpha) \circ T}\, \Downarrow$$

- There is a concept of active parameters of which only one can occur in a super sequent at a time.
- The number of passive parameters (eigenvariables of the indexing sort) is unrestricted.
- This allows the formalization of P-schema (**PA** Schema). Note that no inferences can be applied to the super antecedent.

A Super Sequent Proof



$$\mathcal{E} = \{P(s(n)) = Q(n) \vee Q(s(n)), P(0) = \top, Q(s(n)) = P(n) \wedge Q(n), Q(0) = \top\}$$

Conclusion & Open problems

- Using super sequents for automated deduction.
- Further investigation of the properties of the calculus.
- Further investigation into the relationship between proof Schema and cyclic proofs.

Thank you for your time.