TWO DESIGN QUESTIONS

Alessio Guglielmi

Bath

Proof, Computation, Complexity — Djursholm, 15-19 July 2019

Talk available from my home page and at http://cs.bath.ac.uk/ag/t/TDA.pdf

All about deep inference at http://alessio.guglielmi.name/res/cos

17/7/19
Motivation: Speed-ups for analytic proofs

Corollary of [Aguilera-Baaz, JSL, 2019]

Deep inference has a nonelementary speed-up over cut-free Gentzen proofs of the predicate calculus.
Motivation: Speed-ups for Analytic Proofs

**Corollary** of [Aguilera-Baaz, JSL, 2019]

Deep inference has a nonelementary speed-up over cut-free Gentzen proofs of the predicate calculus.

\[\vdash P_z, \quad \overline{P_z}\]
\[\vdash P_z, \quad \overline{P_z}, Py\]
\[\vdash P_x, P_z, \quad \overline{P_z}, Py\]
\[\vdash P_x, P_z, \quad \overline{P_z} \lor Py\]
\[\vdash P_x \lor P_z, \quad \overline{P_z} \lor Py\]
\[\vdash P_x \lor P_z, \quad \forall y.(\overline{P_z} \lor Py)\]
\[\vdash P_x \lor P_z, \quad \exists x.\forall y.(P_x \lor Py)\]
\[\vdash \forall y.(P_x \lor Py), \quad \exists x.\forall y.(P_x \lor Py)\]
\[\vdash \exists x.\forall y.(P_x \lor Py), \quad \exists x.\forall y.(\overline{P_x} \lor Py)\]
Motivation: Speed-ups for analytic proofs

Corollary of [Aguilera-Baaz, JSL, 2019]

Deep inference has a nonelementary speed-up over cut-free Gentzen proofs of the predicate calculus.

\[
\frac{\exists x. (P_x \vee A_y \cdot P_y)}{
\exists x. (P_x \vee A_y \cdot P_y)
}
\]

Deep inference does not.

Gentzen bureaucracy demands contractions.
Motivation: Speed-ups for analytic proofs

Corollary of [Aguilera-Baaz, JSL, 2019]
Deep inference has a nonelementary speed-up over cut-free Gentzen proofs of the predicate calculus.

\[ \exists x. P_x \lor \forall y. P_y \]

Success is due to quantifier shifts.

Is there a good reason to do without quantifier shifts?
Motivation: speed-ups for analytic proofs

**Corollary** of [Aguilera-Baaz, JSL, 2019]

Deep inference has a nonelementary speed-up over cut-free Gentzen proofs of the predicate calculus.

\[
\frac{\xi}{\exists x. \bar{P}x \lor \forall y. P_y}
\]

Success is due to quantifier shifts.

\[
\frac{\exists x. \bar{P}x \lor \forall y. P_y}{\forall y. (\bar{P}x \lor P_y)}
\]

The speed-up for analytic propositional proofs is exponential.

[Bruscoli-Cuglielmi, ACM ToCL, 2009]

Is there a good reason to do without quantifier shifts?
Motivation: Speed-ups for Analytic Proofs

Corollary of [Aguilera-Baaz, JSL, 2019]

Deep inference has a nonelementary speed-up over cut-free Gentzen proofs of the predicate calculus.

\[
\begin{array}{c}
\exists x. \overline{P_x} \lor \forall y. P_y \\
\exists x. \overline{\exists x. (P_x \lor P_y)}
\end{array}
\]

Success is due to quantifier shifts.

Theorems

The speed-up for analytic propositional proofs is exponential.
[Bruscoli-Cuglielmi, ACAL ToCL, 2009]

Cut-elimination for propositional classical logic is quasipolynomial.
[Jeřábek, JLC, 2009]

Is there a good reason to do without quantifier shifts?
Motivation: canonical forms for proof semantics

Cut elimination for propositional classical logic:

proof with cuts \rightarrow \text{proof without cuts}
Motivation: canonical forms for proof semantics

Cut elimination for propositional classical logic, in Gentzen:

\[ \frac{\frac{\pi_1 \vdash B}{\vdash B, a} \quad \frac{\pi_2 \vdash B}{\vdash B, \overline{a}}}{\frac{\vdash \overline{B}, a \quad \vdash \overline{B}, \overline{a}}{\vdash \overline{B}, \overline{B}}} \]

\[ \frac{\pi_1}{\vdash B} \]

\[ \frac{t}{B} \]
Motivation: canonical forms for proof semantics

Cut elimination for propositional classical logic, in Gentzen:

\[
\begin{align*}
\Pi_1 & \vdash B \\
\Pi_2 & \vdash B \\
B, a & \vdash B, a \\
B, \overline{a} & \vdash B, \overline{a} \\
B & \vdash B, B
\end{align*}
\]

No canonical form, therefore bad semantics.
Motivation: canonical forms for proof semantics

Cut elimination for propositional classical logic, in deep inference:

\[ a \land \neg a \]
\[ \frac{a}{f} \]

*Proofs can be composed by any connective.*
Motivation: canonical forms for proof semantics

Cut elimination for propositional classical logic, in deep inference:

\[ \text{\( \pi \)} \]

\[ \text{\( \pi \)} \]

\[ \text{\( \pi \)} \]

No as in cuts.
Motivation: canonical forms for proof semantics

Cut elimination for propositional classical logic, in deep inference:

\[
\begin{array}{c}
\text{t} \\
\hline
\text{f}
\end{array}
\]

\[
\begin{array}{c}
a \wedge \overline{a} \\
\hline
f
\end{array}
\]

\[
\begin{array}{c}
a \\
\hline
[f \rightarrow \overline{a}]p
\end{array}
\]

\[
\begin{array}{c}
\overline{a} \\
\hline
[f \rightarrow a]p
\end{array}
\]

No as in cuts in both proofs.
Motivation: canonical forms for proof semantics

Cut elimination for propositional classical logic, in deep inference:

\[
\begin{array}{c}
t \\ \pi \\
\hline
f \\
\hline
a \land \bar{a}
\end{array}
\]

\[
\begin{array}{c}
a \\
\hline
[f \rightarrow \bar{a}] \pi
\end{array}
\]

\[
\begin{array}{c}
\bar{a} \\
\hline
[f \rightarrow a] \pi
\end{array}
\]

\[
\begin{array}{c}
t \\
\hline
B
\end{array}
\]

\[
\begin{array}{c}
\bar{a} \\
\hline
B
\end{array}
\]

\[
\begin{array}{c}
a \\
\hline
B
\end{array}
\]

\[
\begin{array}{c}
[f \rightarrow \bar{a}] \pi \\
\hline
B
\end{array}
\]

\[\text{identity} \]

\[\text{disjunction of proofs!} \]

\[\text{contraction} \]

\[\text{No as in cuts.} \]
Motivation: canonical forms for proof semantics

Cut elimination for propositional classical logic, in deep inference:

No cuts. Canonical modulo associativity and commutativity.
Motivation: canonical forms for proof semantics

Cut elimination for propositional classical logic, in deep inference. Example:
Motivation: canonical forms for proof semantics

Cut elimination for propositional classical logic, in deep inference. Example:

Propagate (co)weakenings. Still canonical.
Motivation: Express logics that Gentzen's theory cannot express

Example: $BV = MLL + @$, where $@$ is self-dual non-commutative.

- no interaction $\otimes$
- $\otimes$
- $\otimes$
- $\otimes$
- sequential composition $\triangleleft$
- $\triangleleft$
- $\triangleleft$
- parallel composition = interaction $\triangleright$
Motivation: express logics that Gentzen's theory cannot express.

Example: $\mathcal{B}V = \text{MLL} + \otimes$, where $\otimes$ is self-dual non-commutative.

- **no interaction**
- **sequential composition**
- **parallel composition = interaction**
Motivation: Express logics that Gentzen’s theory cannot express.

Example: $BV = MLL + 	riangle$, where $\triangle$ is self-dual non-commutative.

Question: Is there a proof system that generates all the intermediate formulae between $A \otimes B \to \cdots \to A \otimes B$?
Motivation: express logics that Gentzen's theory cannot express.

Example: $BV = MLL + \Box$, where $\Box$ is self-dual non-commutative.

Question: Is there a proof system that generates all the intermediate formulae between $A \otimes B \rightarrow \cdots \rightarrow A \boxtimes B$?

Yes:

\[
\frac{(A \otimes B) \otimes (C \otimes D)}{(A \otimes C) \otimes (B \otimes D)}
\]

\[
\frac{(A \otimes B) \otimes (C \bowtie D)}{(A \otimes C) \bowtie (B \otimes D)}
\]

\[
\frac{(A \bowtie B) \otimes (C \bowtie D)}{(A \bowtie C) \otimes (B \bowtie D)}
\]

\[
\frac{(A \otimes C) \bowtie (B \bowtie D)}{(A \bowtie C) \otimes (B \bowtie D)}
\]

+ commutativity (same shape)

and mirror images

This follows from a combinatorial argument via relation webs [Cuglielmi, ACM ToCL, 2007]. It can be generalised to linear logics with an arbitrary number of relations.
Motivation: express logics that Gentzen's theory cannot express.

Example: $BV = MLL + \Box$, where $\Box$ is self-dual non-commutative.

Question: Is there a proof system that generates all the intermediate formulae between $A \otimes B \rightarrow \cdots \rightarrow A \otimes B$?

Yes:

\[
\begin{align*}
(A \otimes B) \otimes (C \otimes D) & \quad \frac{}{(A \otimes C) \otimes (B \otimes D)} \\
(A \otimes C) \otimes (B \otimes D) & \quad (A \otimes B) \otimes (C \otimes D) \\
(A \otimes B) \otimes (C \otimes D) & \quad (A \otimes C) \otimes (B \otimes D) \\
(A \otimes C) \otimes (B \otimes D) & \quad (A \otimes B) \otimes (C \otimes D) \\
(A \otimes C) \otimes (B \otimes D) & \quad (A \otimes B) \otimes (C \otimes D)
\end{align*}
\]

+ commutativity (same shape) and mirror images

Theorems: There is an analytic system for $BV$ in deep inference [Guglielmi, ACM TOCL 2007] but not in Gentzen [Tiu, LNCS, 2006]. Applications in process algebras (many papers, see web).
Motivation: express logics that Gentzen’s theory cannot express.

Example: \( BV = \text{MLL} + \alpha \), where \( \alpha \) is self-dual non-commutative.

Question: Is there a proof system that generates all the intermediate formulae between \( A \otimes B \rightarrow \cdots \rightarrow A \& B \)?

Yes:

\[
\frac{(A \rightarrow B) \& (C \rightarrow D)}{(A \otimes C) \& (B \otimes D)} \quad \text{\textquoteleft satturates up\textquoteright} \quad \frac{\alpha \beta}{(A \& C) \beta (B \& D)}
\]

This shape generates the rules:

\[
\frac{(A \& B) \rightarrow (C \& D)}{(A \otimes C) \& (B \otimes D)}
\]

Theorems: There is an analytic system for \( BV \) in deep inference [Guglielmi, ACM TOCL 2007] but not in Gentzen [Tiu, LNCS, 2006]. Applications in process algebras (many papers, see web).
Motivation: express logics that Gentzen's theory cannot express.

Example: $BV = MLL + \Diamond$, where $\Diamond$ is self-dual non-commutative.

Question: Is there a proof system that generates all the intermediate formulae between $A \otimes B \rightarrow \cdots \rightarrow A \otimes B$?

Yes:

\[
\begin{align*}
\otimes & \quad \vdash (A \otimes B) \otimes (C \otimes D) \\
\otimes \otimes & \quad \vdash (A \otimes C) \otimes (B \otimes D) \\
\end{align*}
\]

'saturates up' \hspace{1cm} \vdash = \vdash = \vdash

\[
\begin{align*}
\beta & \quad \vdash (A \bowtie B) \bowtie (C \bowtie D) \\
\beta \bowtie & \quad \vdash (A \bowtie C) \bowtie (B \bowtie D) \\
\end{align*}
\]

'saturates down' \hspace{1cm} \bowtie = \bowtie = \bowtie

This shape generates the rules.

\[
\begin{align*}
\beta & \quad \vdash (A \bowtie B) \bowtie (C \bowtie D) \\
\beta \bowtie & \quad \vdash (A \bowtie C) \bowtie (B \bowtie D) \\
\end{align*}
\]

'saturates down'

Theorems: There is an analytic system for $BV$ in deep inference [Guguelmi, ACM Tocl 2007] but not in Gentzen [Tiu, LNCS, 2006]. Applications in process algebras (many papers, see web).
Motivation: express logics that Gentzen's theory cannot express.

Example: $\text{BV} = \text{MLL} + \Delta$, where $\Delta$ is self-dual non-commutative.

Question: Is there a proof system that generates all the intermediate formulae between $A \otimes B \to \cdots \to A \& B$?

Yes:

\[
\begin{align*}
\frac{\Delta}{A \otimes B \otimes (C \& D)} & \quad \frac{\Delta}{(A \otimes C) \& (B \& D)} \\
\frac{(A \& B) \otimes (C \& D)}{(A \otimes C) \& (B \& D)} & \quad \frac{(A \& B) \otimes (C \& D)}{(A \otimes C) \& (B \& D)} \quad \sim \quad \Delta
\end{align*}
\]

This shape generates the rules.

Theorems: There is an analytic system for $\text{BV}$ in deep inference [Guglielmi, ACM-TOCL 2007] but not in Gentzen [Tiu, LMCS, 2006]. Applications in process algebras (many papers, see web).
Motivation: Express logics that Gentzen's theory cannot express.

Example: $BV = MLL + \land$, where $\land$ is self-dual non-commutative.

Question: Is there a proof system that generates all the intermediate formulae between $A \land B \to \cdots \to A \lor B$?

Yes:

\begin{align*}
\hat{\land} & \quad \hat{\land} (A \land B) \land (C \land D) \\
& \quad \hat{\land} (A \land C) \land (B \land D) \\
\hat{\lor} & \quad \hat{\lor} (A \lor B) \lor (C \lor D) \\
& \quad \hat{\lor} (A \lor C) \lor (B \lor D)
\end{align*}

This shape generates the rules.

Saturation can happen on the left, too, of course.
The generating shape

\[
\begin{align*}
\alpha \beta & \quad (A \beta B) \alpha (C \hat{\beta} D) \\
\alpha \hat{\beta} & \quad (A \alpha C) \beta (B \alpha D)
\end{align*}
\]
The generating shape

**Binary - Binary**

\[
\begin{align*}
\wedge \quad (A \wedge B) \wedge (C \wedge D) \\
\wedge \quad (A \wedge C) \wedge (B \wedge D) \\
\wedge \quad (A \wedge C) \vee (B \wedge D) \\
\wedge \quad (A \wedge C) \wedge (B \wedge D) \\
\wedge \quad (A \wedge C) \vee (B \wedge D) \\
\end{align*}
\]

**Examples**

- Commutativity / Associativity
- Switch (classical logic)
- Switch (linear logic)
- Medial
The generating shape

**examples**

\[ \wedge \wedge (A \wedge B) \land (C \wedge D) \]
\[ (A \wedge C) \land (B \wedge D) \]

\[ \wedge \lor (A \lor B) \land (C \lor D) \]
\[ (A \lor C) \lor (B \lor D) \]

\[ \otimes \otimes (A \otimes B) \otimes (C \otimes D) \]
\[ (A \otimes C) \otimes (B \otimes D) \]

\[ \lor \lor (A \lor B) \lor (C \lor D) \]
\[ (A \lor C) \land (B \lor D) \]

**binary-unary**

\[ \alpha \beta (\beta A) \alpha (\hat{A} B) \]
\[ \beta (A \alpha B) \]

\[ (A \beta B) \alpha (C \hat{\beta} D) \]
\[ (A \alpha C) \beta (B \alpha D) \]
The generating shape

\[ \begin{align*}
\text{binary-binary} & \\
\alpha \beta & \frac{(A \land B) \land (C \land D)}{(A \land C) \land (B \land D)} \\
\land & \frac{(A \lor B) \land (C \land D)}{(A \land C) \lor (B \land D)} \\
\lor & \frac{(A \land B) \lor (C \land D)}{(A \land C) \land (B \lor D)} \\
\otimes & \frac{(A \otimes B) \otimes (C \otimes D)}{(A \otimes C) \otimes (B \otimes D)} \\
\text{vector} & \frac{(A \land B) \lor (C \land D)}{(A \lor C) \land (B \lor D)}
\end{align*} \]

\[ \begin{align*}
\text{binary-unary} & \\
\alpha \beta & \frac{(\beta A) \alpha (\hat{A} B)}{(\beta A \land B)} \\
\land & \frac{(A \land B) \land (A \land B)}{(A \land B)} \\
\lor & \frac{(A \lor B) \lor (A \lor B)}{(A \lor B)} \\
\otimes & \frac{(A \otimes B) \otimes (A \otimes B)}{(A \otimes B)} \\
\text{vector} & \frac{(A \lor B) \lor (A \lor B)}{(A \lor B)}
\end{align*} \]

\[ \begin{align*}
\text{examples} & \\
\hat{A} & \frac{\Diamond A \land \Box B}{\Diamond (A \land B)} \\
\otimes & \frac{\Box A \otimes \Diamond B}{\Box (A \otimes B)} \\
\exists x & \frac{\exists x (A \lor B)}{\exists x (A \land B)} \\
\text{co-k} & \frac{\Box A \otimes \Diamond B}{\Box (A \otimes B)} \\
\text{co-promotion} & \frac{\exists x (A \lor B)}{\exists x (A \land B)} \\
\text{quantifier} & \frac{\exists x (A \land B)}{\exists x (A \lor B)} \\
\text{shift} & \frac{\exists x (A \land B)}{\exists x (A \lor B)}
\end{align*} \]
The generating shape

**examples**

\[ \alpha \beta \]
\[ (A \land B) \land (C \land D) \]
\[ \land \land \]
\[ (A \land C) \land (B \land D) \]
\[ \beta (A \land B) \]

**binary-binary**

\[ \alpha \beta \]
\[ (A \land B) \land (C \land D) \]
\[ \land \land \]
\[ (A \land C) \land (B \land D) \]
\[ \beta (A \land B) \]

**binary-unary**

\[ \alpha \beta \]
\[ (A \land B) \land (C \land D) \]
\[ \land \land \]
\[ (A \land C) \land (B \land D) \]
\[ \beta (A \land B) \]

**unary-unary**

\[ \alpha \beta \]
\[ (A \land B) \land (C \land D) \]
\[ \land \land \]
\[ (A \land C) \land (B \land D) \]
\[ \beta (A \land B) \]

**examples**

\[ \land \land \]
\[ (A \land B) \land (C \land D) \]
\[ \land \land \]
\[ (A \land C) \land (B \land D) \]
\[ \beta (A \land B) \]

\[ \land \land \]
\[ (A \land B) \land (C \land D) \]
\[ \land \land \]
\[ (A \land C) \land (B \land D) \]
\[ \beta (A \land B) \]

\[ \land \land \]
\[ (A \land B) \land (C \land D) \]
\[ \land \land \]
\[ (A \land C) \land (B \land D) \]
\[ \beta (A \land B) \]

\[ \land \land \]
\[ (A \land B) \land (C \land D) \]
\[ \land \land \]
\[ (A \land C) \land (B \land D) \]
\[ \beta (A \land B) \]

\[ \land \land \]
\[ (A \land B) \land (C \land D) \]
\[ \land \land \]
\[ (A \land C) \land (B \land D) \]
\[ \beta (A \land B) \]
The generating shape — **Surprise! Non-linear rules are generated**

\[
\begin{align*}
\alpha &\vdash (A \land B) \land (C \land D) \\
\Rightarrow &\vdash (A \land C) \land (B \land D) \\
\wedge &\vdash (f \land t) \land (t \land f) \\
\wedge &\vdash (f \land t) \land (t \land f) \\
\lor &\vdash (f \lor t) \lor (f \lor t) \\
\lor &\vdash (f \lor t) \lor (t \lor t)
\end{align*}
\]
The generating shape - Surprise! Non-linear rules are generated

\[ \alpha \beta \]

\[ (A \beta B) \alpha (C \beta D) \]

\[ (A \alpha C) \beta (B \alpha D) \]

\[ \wedge a \]

\[ (f \land t) \land (t \land f) \]

\[ \frac{a \land \overline{a}}{f} \]

if we take

\[ \begin{cases} 
  f \land f = f \\
  f \land t = a \\
  t \land a = \overline{a} \\
  t \land t = t 
\end{cases} \]

i.e., atoms are superpositions of truth values.
The generating shape — Surprise! Non-linear rules are generated

\[
\begin{align*}
\alpha \beta & \quad \frac{(A \beta B) \alpha (C \beta D)}{(A \alpha C) (B \alpha D)} \\
\wedge a & \quad \frac{(f \land t) \land (t \land f)}{(f \land t) \land (t \land f)} = \frac{a \land \bar{a}}{f} \\
\vee a & \quad \frac{(f \lor t) \lor (t \lor f)}{(f \lor t) \lor (t \lor f)} = \frac{a \lor a}{a}
\end{align*}
\]

if we take

\[
\begin{align*}
f \land f &= f \\
f \land t &= a \\
t \land f &= \bar{a} \\
t \land t &= t
\end{align*}
\]

i.e., atoms are superpositions of truth values.
The generating shape - surprise! Non-linear rules are generated.

Except for unit equations, every rule for all mainstream logics gets generated.

This yields a uniform and general normalisation theory.

\[ \alpha \beta \]

\[ (A \land B) \land (C \land D) \]

\[ \land a \]

\[ (f \land t) a (t \land f) \]

\[ = \]

\[ a \land \bar{a} \]

\[ f \]

\[ \text{if we take} \]

\[ \begin{align*}
  f \land f &= f \\
  f \land t &= a \\
  t \land f &= \bar{a} \\
  t \land t &= t
\end{align*} \]

i.e., atoms are superpositions of truth values.

\[ \lor a \]

\[ (f \lor f) a (t \lor t) \]

\[ = \]

\[ a \lor a \]

\[ a \]
The generating shape — Surprise! Non-linear rules are generated.

Except for unit equations, every rule for all mainstream logics gets generated.

This yields a uniform and general normalisation theory
[aler Tubella-Guglielmi, ACM ToCL, 2018 + papers in preparation].

wild = atoms inside atoms
The generating shape - surprise! Non-linear rules are generated.

Except for unit equations, every rule for all mainstream logics gets generated.

Work in progress - very technical, almost done.

This yields a uniform and general normalisation theory [ALER TUBELLA-GUGLIELMI, ACM ToCL, 2018 + papers in preparation].

wild = atoms inside atoms
Question 1 Why is the shape so successful?
Questions

Question 1 Why is the shape so successful?

Question 2 Does designing a proof system around the equation 
\[(\tau \rightarrow x)B \ast (\nu \rightarrow x)B = [(\tau \ast \nu) \rightarrow x]B, \text{ where } x \in \{v, \lambda, \ldots\}, \text{ make sense?}\]
Questions

Question 1
Why is the shape so successful?

Question 2
Does designing a proof system around the equation 
\([\tau \rightarrow x]B \ast [u \rightarrow x]B = [(\tau \ast u) \rightarrow x]B\), where \(x \in \{v, \lambda, \ldots\}\), make sense?

Check the next example with induction and note:

\[
\begin{array}{c}
\begin{array}{c}
\ast \ \ \ \ [\{} \rightarrow x\}B
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
[\tau \rightarrow x]B \ast [u \rightarrow x]B
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
[(\tau \ast u) \rightarrow x]B
\end{array}
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\ast \ \ \ \ [\{} \rightarrow x\}B
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
[(\tau \ast u) \rightarrow x]B
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
[\tau \rightarrow x]B \ast [u \rightarrow x]B
\end{array}
\end{array}
\end{array}
\]
Example with induction

AO \rightarrow \text{formula } Ax \text{ where every free } x \text{ is substituted with } 0
Example with Induction

\[ \top \quad \text{proof of } A_0 \text{ in deep inference} \]
Example with induction

\[ \forall n. (A_n \land C_{A_n}) \]

Proof of \( \forall n. (A_n \land C_{A_n}) \), where \( s \) stands for the successor function.
Example with induction

\[ \Pi \]
\[ A_0 \]

\[ [O \rightarrow \pi] A_\pi \]

\[ \forall n \quad \Pi \quad A_n \quad \Pi \quad A_{sn} \]

\[ \text{[O} \rightarrow \pi \text{]} \text{ is an indicated (formal) substitution} \]
Example with Induction

\[ \Pi \]
\[ A_0 \]

\[ \overset{0 \rightarrow x}{\Rightarrow} A_x \]

\[ \forall n, \Pi \]
\[ A_n \]
\[ \overset{n \rightarrow x}{\Rightarrow} A_x \]
\[ \overset{sn \rightarrow x}{\Rightarrow} A_x \]

Applying equational inference steps derived from equation:

\[ [\xi \rightarrow x]B = [\xi \rightarrow x]B \]

Where \([\xi \rightarrow x]\) is an actual substitution.
Example with Induction

\[
\prod
\begin{array}{c}
A_0 \\
[0 \rightarrow x] A_x
\end{array}
\wedge
\prod
\begin{array}{c}
\forall n \quad A_n \\
[n \rightarrow x] A_x
\end{array}
\subseteq
\begin{array}{c}
A_{sn} \\
[sn \rightarrow x] A_x
\end{array}
\]
Example with Induction

\[ \Pi \]
\[ A_0 \]
\[ \text{[O} \rightarrow x]\text{Ax} \]
\[ \land \]
\[ \text{[n} \rightarrow x]\text{Ax} \]
\[ \Rightarrow \]
\[ \forall n \text{. } (n \supset \text{sn}) \]

Applying an equational inference step derived from equations:

\[ [\tau \rightarrow x]B \times [\nu \rightarrow x]B = [(\tau \times \nu) \rightarrow x]B \]

And

\[ Q_y([\tau \rightarrow x]B) = [Q_y\tau \rightarrow x]B, \]

where \( \times \) is any connective, \( Q \) is any quantifier and \( y \) is not free in \( B \).
Example with Induction

\[ \prod \]
\[ A_0 \]
\[ \bigvee \]
\[ [O \rightarrow x] Ax \]

\[ \prod \]
\[ A_n \rightarrow A_{sn} \]
\[ \forall n. \bigvee \]
\[ \bigvee \]
\[ [h \rightarrow x] Ax \]
\[ [sn \rightarrow x] Ax \]

\[ \forall n. (u \sqcup sn) \]

\[ \bigvee \]
\[ 0 \bigvee \]
\[ \bigvee \]
\[ \rightarrow x Ax \]

Applying an equational inference step derived from equations:

\[ [\tau \rightarrow x] B \ast [u \rightarrow x] B = [(\tau \ast u) \rightarrow x] B \]

and

\[ Q_{y}([\tau \rightarrow x] B) = [Q_{y} \tau \rightarrow x] B, \]

where \( \ast \) is any connective, \( Q \) is any quantifier and \( y \) is not free in \( B \).

Could this provide a speed-up?
Example with induction

\[
\frac{\Pi}{A_0} \quad \land \\
[0 \to x] A_x
\]

\[
\frac{\Pi}{A_n} \quad \land \\
[0 \to x] A_x \quad \land \\
[sn \to x] A_x
\]

\[
\frac{\forall n. (u \subseteq s_n)}{\forall n. (u \subseteq s_n) \land \forall n. (u \subseteq s_n) \land \forall n. (u \subseteq s_n)}
\]

\[
\rightarrow x A_x
\]

Cocoutraction
Example with Induction

\[\prod\]
\[A \vdash 0\]
\[\left[ O \rightarrow x \right] Ax\]

\[\prod\]
\[\forall n, 0 \vdash A_n \land 4s_n \rightarrow A_{sn}\]
\[\left[ n \rightarrow x \right] Ax \land \left[ sn \rightarrow x \right] 4s_n\]

\[\left[ O \rightarrow p, sO \rightarrow q \right]\]
\[\left[ O \rightarrow p, sO \rightarrow q \right]\]
\[\forall n, 0 \vdash (n \in s_n) \land \forall n, (n \in s_n) \land \forall n, (n \in s_n)\]
\[\left[ O \vdash x \right] Ax\]

p and q are not free in \(O \vdash \forall n, (n \in s_n)\), so nothing changes
Example with Induction

\[ \prod \]
\[ A \vdash \exists x \neg A \]
\[ [\text{for } n \in \text{nat}] A \vdash [\text{for } n \in \text{nat}] \neg A \]
\[ \forall n. [\text{for } n \in \text{nat}] A \vdash [\text{for } n \in \text{nat}] \neg A \]

Applying the rule
\[ \forall y. B \]
\[ B \]

(quantifiers only provide scope, not witnesses)
Example with Induction

\[
\prod \frac{A_0}{[0 \to x] A_x} \land
\prod \frac{\forall n. (A_n \land A_{sn})}{[n \to x] A_x \land [sn \to x] A_x}
\]

\[
\prod \frac{\forall n. (u \subset s n)}{p \subset s p \land \forall n. (u \subset s n) \land q \subset s q}
\]

\[
\forall \land (\forall \subset s 0) \land (s 0 \subset c s 0)
\]

Applying
\[
[t \to x] B = [t \to x] B
\] (already seen)
Example with Induction

\[ \Pi \]
\[ A_0 \]
\[ [0 \rightarrow x] A_x \]

\[ \Pi \]
\[ \forall n. [n \rightarrow x] A_x \]
\[ [sn \rightarrow x] A_x \]
\[ \forall n. (n \in sn) \]
\[ \forall n. (n \in sn) \]
\[ \forall n. (n \in sn) \]
\[ p \subseteq sp \]
\[ q \subseteq sq \]

\[ 0 \wedge (0 \subseteq s0) \wedge (s0 \subseteq ss0) \]

\[ 0 \wedge 0 \]
\[ s0 \wedge s0 \]
\[ s0 \wedge ss0 \]
\[ f \]
\[ f \]
\[ ss0 \]

Open deduction derivation of switch instances — note negation on terms
Example with Induction

\[ \begin{align*}
\Pi \\
A_0 \\
\hline
[O \rightarrow x] A_x \\
\wedge \\
\forall n. \frac{A_n \land [n \rightarrow x] A_x \land [sn \rightarrow x] A_x}{A_{sn}}
\end{align*} \]

\[ \begin{align*}
[O \rightarrow p, \\
sO \rightarrow q] \\
\wedge \\
\forall n. (u \subseteq sn) \\
\bar{c} \\
\forall n. (u \subseteq sn) \land (v \subseteq sn) \\
\Rightarrow \bar{c}
\end{align*} \]

\[ O \land (O \subseteq sO) \land (sO \subseteq ssO) \]

\[ \frac{0 \land \bar{0} \lor \bar{0} \land ssO}{ssO} \]

applying
\[ [t \rightarrow x] B = [t \Rightarrow x] B \]
(already seen)
Example with Induction

\[
\begin{align*}
\Pi & \quad A_0 \\
\wedge & \quad \left[ O \rightarrow x \right] A_x \\
\end{align*}
\]

\[
\begin{align*}
\Pi & \quad A_n \\
\wedge & \quad \left[ n \rightarrow x \right] A_x \\
\forall n & \quad \left[ n \rightarrow x \right] A_x \\
\forall n & \quad \left[ s_n \rightarrow x \right] A_x \\
\end{align*}
\]

\[
\begin{align*}
\forall n (u \subseteq s_n) \\
\wedge & \quad \neg \exists c \\
\forall n (u \subseteq s_n) & \quad \forall n (u \subseteq s_n) \\
\wedge & \quad p \subseteq s_p \\
\wedge & \quad q \subseteq s_q \\
\rightarrow x & \quad A_x \\
\end{align*}
\]

\[
\begin{align*}
O & \quad O \subseteq s_0 \\
(0 \subseteq s_0) & \quad (s_0 \subseteq ss_0) \\
\end{align*}
\]

\[
\begin{align*}
O \wedge O \overline{O} & \quad \vee \overline{i} \\
\vee i & \quad \vee s_0 \wedge \overline{s_0} \\
\overline{i} & \quad \overline{f} \\
\vee s_0 & \quad \vee ss_0 \\
\overline{f} & \quad \overline{f} \\
ss_0 & \quad ss_0 \\
\end{align*}
\]

\[
\begin{align*}
\text{Ass}_0
\end{align*}
\]

cuts can be eliminated as usual
Example with Induction

\[
\Pi
\frac{A_0}{O \rightarrow \pi A_0}
\]
\[
\Pi
\frac{A_n}{\forall n \rightarrow \pi A_n}
\]
\[
\frac{A_{sn}}{\forall n \rightarrow \pi A_{sn}}
\]

\[
\forall n. (n \subset s n)
\]
\[
\frac{c}{\forall n. (n \subset s n) \wedge (n \subset s n) \wedge \overline{s n}}
\]
\[
\frac{\overline{c} \wedge s n \wedge s n}{s s 0}
\]

\[
\frac{s s 0 \rightarrow n}{A_{s s 0}}
\]

we can also use an induction rule instead

\[
\forall n. (n \subset s n)
\]
\[
\frac{0 \wedge \forall n. (n \subset s n)}{s s 0}
\]
Could something like this lead to better quantification?

A proof theory for the $\xi$-calculus?