A Proposal for a New Foundation for Proof Theory

Alessio Guglielmi

University of Bath

11 March 2016

This talk is available at `http://cs.bath.ac.uk/ag/t/PNFPT.pdf`
Deep inference web site: `http://alessio.guglielmi.name/res/cos/`

Outline

Proof semantics – When are two given proofs the same?

Deep inference – Free composition of proofs

Deep inference for classical logic

Cut elimination

The noncommutative linear logic BV

Atomic flows – Locality yields topology

Normalisation with atomic flows – Topology is enough to normalise

When are two given proofs the same?

Remove bureaucracy; compare shapes. Proofs $\to$ proof nets:

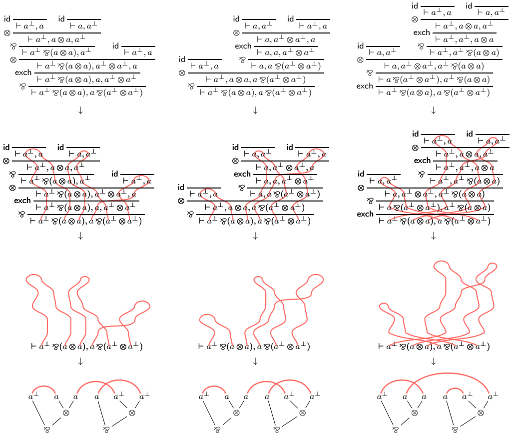


Figure taken from [20]

Remove too much $\to$ no more a proof system: proof nets are not a proof system.

Proof Systems

- Proof system = algorithm checking proofs in polytime.
- Theorem (Cook and Reckhow):

$$\exists \textit{ super proof system}$$
$$\textit{iff}$$
$$\mathrm{NP} = \text{co-NP}$$

where

super = with polysize proofs over each proved tautology

Compressing proofs

How can we make proofs smaller? Known proof theoretic mechanisms:

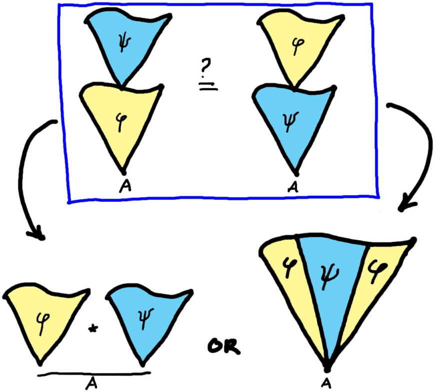
1. Re-use the same sub-proof: cut rule. Proof theory.
2. Re-use the same sub-proof: dagness, or cocontraction: $\mathsf{c}\!\uparrow \dfrac{A}{A \wedge A}$.
3. Substitution: $\mathsf{sub}\,\dfrac{A}{A\sigma}$. In Frege, equivalent to (4).
4. Tseitin extension: $p \leftrightarrow A$ (where p is a fresh atom). Optimal?
5. Higher orders (including 2$^{\text{nd}}$ order propositional).

1–4 (and a bit also 5) have a lot to do with proof composition – our main tool.

Our main objective is providing for small spaces of canonical proofs (= eliminating bureaucracy = getting good proof semantics).

Compressing proof (search) spaces

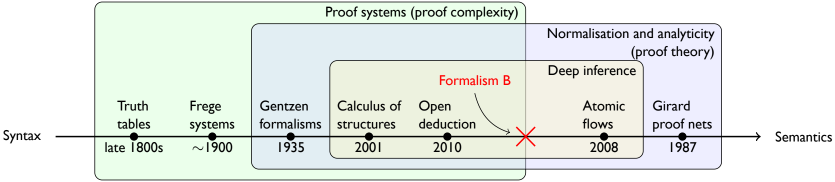
How can we make proof (search) spaces smaller? This also has a lot to do with proof composition:



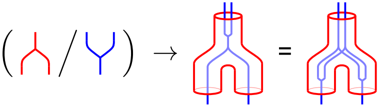
By allowing for more composed proofs we get:

- More proofs in the proof (search) space. This might be bad.
- Small (search) subspaces of canonical proofs. This is good.

Towards proof systems closest to semantics



- Open deduction: established deep inference – full normalisation theory.
- 'Formalism B': getting the power of Frege + substitution (possibly optimal proof system) by incorporating substitution, guided by the geometry of atomic flows:



What is deep inference?

It's the free composition of proofs via the same connectives as formulae.

If

$$\Phi = \begin{array}{c} A \\ \| \\ B \end{array} \qquad \text{and} \qquad \Psi = \begin{array}{c} C \\ \| \\ D \end{array}$$

are two proofs with, respectively, premisses A and C and conclusions B and D, then

$$(\Phi \wedge \Psi) = \begin{array}{c} (A \wedge C) \\ \| \\ (B \wedge D) \end{array} \qquad \text{and} \qquad [\Phi \vee \Psi] = \begin{array}{c} [A \vee C] \\ \| \\ [B \vee D] \end{array}$$

are valid proofs with, respectively, premisses $(A \wedge C)$ and $[A \vee C]$, and conclusions $(B \wedge D)$ and $[B \vee D]$.

Why deep inference?

- To recover a De Morgan premiss-conclusion symmetry that is lost in Gentzen [2].
- To obtain new notions of normalisation in addition to cut elimination [11, 10].
- To shorten analytic proofs by exponential factors compared to Gentzen [6, 8].
- To obtain quasipolynomial-time normalisation for propositional logic [7].
- To express logics that cannot be expressed in Gentzen [22, 3].
- To make the proof theory of a vast range of logics regular and modular [3].
- To get proof systems whose inference rules are local, which is usually impossible in Gentzen [19].

Why deep inference? (cont.)

- To inspire a new generation of proof nets and semantics of proofs [21].
- To investigate the nature of cut elimination [10, 12].
- To type optimal versions of the λ-calculus that are not typeable in Gentzen [13, 14].
- To model process algebras [5, 16, 17, 18].
- To model quantum causal evolution [1] …
- … and much more.

Why deep inference? (cont.)

Several formalisms can be designed in deep inference: Calculus of Structures (CoS), Nested Sequents, Open Deduction, Formalism B, …

CoS and open deduction are equivalent under any reasonable point of view, so we adopt open deduction. (CoS is convenient for certain technical aspects.)

Nested sequents is not full deep inference.

Formalism B is still in development.

Deep inference system SKS for classical logic

- Atomic/structural rules:

$$\mathsf{i}{\downarrow}\,\frac{\mathsf{t}}{a \vee \bar{a}} \qquad \mathsf{w}{\downarrow}\,\frac{\mathsf{f}}{a} \qquad \mathsf{c}{\downarrow}\,\frac{a \vee a}{a}$$

identity   *weakening*   *contraction*

$$\mathsf{i}{\uparrow}\,\frac{a \wedge \bar{a}}{\mathsf{f}} \qquad \mathsf{w}{\uparrow}\,\frac{a}{\mathsf{t}} \qquad \mathsf{c}{\uparrow}\,\frac{a}{a \wedge a}$$

cut   *coweakening*   *cocontraction*

- Linear/logical rules:

$$\mathsf{s}\,\frac{A \wedge [B \vee C]}{(A \wedge B) \vee C} \qquad \mathsf{m}\,\frac{(A \wedge B) \vee (C \wedge D)}{[A \vee C] \wedge [B \vee D]}$$

switch   *medial*

- Plus an '=' linear rule (associativity, commutativity, units).
- Negation on atoms only.

The cut is atomic.

SKS is complete for propositional logic. See [4].

Example

$$
\begin{array}{c}
[a \vee b] \wedge a \\
\| \\
([a \vee b] \wedge a) \wedge ([a \vee b] \wedge a)
\end{array}
=
\boxed{\mathsf{m}\,\dfrac{\boxed{\mathsf{c}\!\uparrow \dfrac{a}{a \wedge a}} \vee \boxed{\mathsf{c}\!\uparrow \dfrac{b}{b \wedge b}}}{[a \vee b] \wedge [a \vee b]}}
\wedge
\boxed{\mathsf{c}\!\uparrow \dfrac{a}{a \wedge a}}
$$

Structural rules on generic formulae can be replaced by structural rules on atoms.

Example with quantifiers

$$
\mathsf{c}{\downarrow}\,\dfrac{\mathsf{i}{\downarrow}\,\dfrac{\mathsf{t}}{\exists x \forall y \left[\boxed{\mathsf{w}{\downarrow}\,\dfrac{\mathsf{f}}{\overline{p(x)}}} \vee p\left(y\right) \right] \vee \exists x \forall y \left[\overline{p\left(x\right)} \vee \boxed{\mathsf{w}{\downarrow}\,\dfrac{\mathsf{f}}{p\left(y\right)}} \right]}}{\exists x \forall y \left[\overline{p\left(x\right)} \vee p\left(y\right) \right]}
$$

This is more natural than in Gentzen because there is no waste in the proof.

Locality

Deep inference allows for locality,

i.e.,

inference steps can be checked in constant time (so, they are small).

E.g., atomic cocontraction:

$$\mathsf{m}\,\frac{\dfrac{a}{a \wedge a} \vee \dfrac{b}{b \wedge b}}{[a \vee b] \wedge [a \vee b]} \quad \wedge \quad \frac{a}{a \wedge a}$$

Thanks to locality Gundersen, Heijltjes and Parigot obtained a typed λ-calculus that achieves fully lazy sharing [13].

In Gentzen:

- no locality for (co)contraction (counterexample in [2]),
- no local reduction of cut into atomic form.

Reduction of cut to atomic form

Apply repeatedly—and locally:

$$\mathsf{i}{\uparrow}\,\frac{[A \vee B] \wedge (\bar{A} \wedge \bar{B})}{\mathsf{f}} = \mathsf{s}\,\frac{\boxed{\mathsf{s}\,\frac{[A \vee B] \wedge \bar{B}}{A \vee (B \wedge \bar{B})}} \wedge \bar{A}}{\boxed{\mathsf{i}{\uparrow}\,\frac{A \wedge \bar{A}}{\mathsf{f}}} \vee \boxed{\mathsf{i}{\uparrow}\,\frac{B \wedge \bar{B}}{\mathsf{f}}}}$$

Proof complexity does not increase!

Analyticity costs much less (1)

Statman tautologies:

$$
\begin{aligned}
S_1 &= (a \wedge b) \vee \bar{a} \vee \bar{b} \quad , \\
S_2 &= (c \wedge d) \vee \left(\left[\bar{c} \vee \bar{d}\right] \wedge a \wedge \left[\bar{c} \vee \bar{d}\right] \wedge b \right) \vee \bar{a} \vee \bar{b} \quad , \\
S_3 &= (e \wedge f) \vee \left(\left[\bar{e} \vee \bar{f}\right] \wedge c \wedge \left[\bar{e} \vee \bar{f}\right] \wedge d \right) \vee \\
&\quad\ \left(\left[\bar{e} \vee \bar{f}\right] \wedge \left[\bar{c} \vee \bar{d}\right] \wedge a \wedge \left[\bar{e} \vee \bar{f}\right] \wedge \left[\bar{c} \vee \bar{d}\right] \wedge b \right) \vee \bar{a} \vee \bar{b}
\end{aligned}
$$

and so on…

In the cut-free sequent calculus proofs grow exponentially.

Analyticity costs much less (2)

Open deduction proof of S_1:

$$\boxed{(a \wedge b) \vee \bar{a} \vee \bar{b}} \overset{\mathsf{t}}{\Big\Vert}{\scriptstyle \mathsf{i}\downarrow,\mathsf{s}} \quad = \quad \mathsf{s}\,\frac{\boxed{\mathsf{i}\downarrow\,\dfrac{\mathsf{t}}{a \vee \bar{a}}} \wedge \boxed{\mathsf{i}\downarrow\,\dfrac{\mathsf{t}}{b \vee \bar{b}}}}{\boxed{\mathsf{s}\,\dfrac{[a \vee \bar{a}] \wedge b}{(a \wedge b) \vee \bar{a}}} \vee \bar{b}}$$

Analyticity costs much less (3)

Open deduction proof of S_2:

$$
\begin{array}{c}
\mathsf{t} \\
\Vert\, \mathsf{i}{\downarrow},\mathsf{s} \\
\boxed{
\boxed{
\begin{array}{c}
\boxed{
\boxed{
\begin{array}{c}
\mathsf{t} \\
\Vert\, \mathsf{i}{\downarrow},\mathsf{s} \\
(c \wedge d) \vee \bar{c} \vee \bar{d}
\end{array}
}
\wedge a \wedge
\boxed{
\begin{array}{c}
\mathsf{t} \\
\Vert\, \mathsf{i}{\downarrow},\mathsf{s} \\
(c \wedge d) \vee \bar{c} \vee \bar{d}
\end{array}
}
\wedge b
} \\
\Vert\, \mathsf{s} \\
\boxed{
\boxed{
\begin{array}{c}
(c \wedge d) \vee (c \wedge d) \\
\Vert\, \mathsf{c}{\downarrow},\mathsf{m} \\
\boxed{c \wedge d}
\end{array}
}
\vee \left(\left[\bar{c} \vee \bar{d} \right] \wedge a \wedge \left[\bar{c} \vee \bar{d} \right] \wedge b \right)
}
\end{array}
}
\vee \bar{a} \vee \bar{b}
}
\end{array}
$$

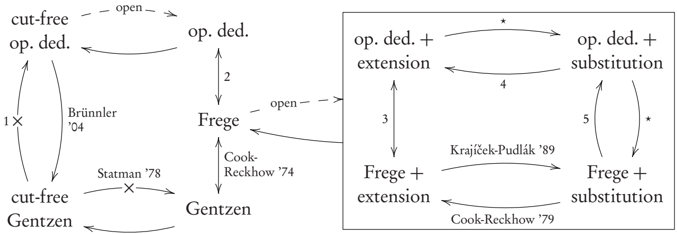
Analyticity costs much less (4)

Open deduction proof of S_3:

$$
\begin{array}{c}
\mathsf{t} \\
\Big\Vert\,{\scriptstyle \mathsf{i}\downarrow,\mathsf{s}} \\
\boxed{
\begin{array}{c}
\boxed{
\boxed{\begin{array}{c} \mathsf{t} \\ \Vert\,{\scriptstyle \mathsf{i}\downarrow,\mathsf{s}} \\ (c \wedge d) \vee \bar{c} \vee \bar{d} \end{array}}
\wedge a \wedge
\boxed{\begin{array}{c} \mathsf{t} \\ \Vert\,{\scriptstyle \mathsf{i}\downarrow,\mathsf{s}} \\ (c \wedge d) \vee \bar{c} \vee \bar{d} \end{array}}
\wedge b
} \\
\Big\Vert\,{\scriptstyle \mathsf{s}} \\
\boxed{
\boxed{\begin{array}{c}
(c \wedge d) \vee (c \wedge d) \\
\Vert\,{\scriptstyle \mathsf{c}\downarrow,\mathsf{m}} \\
\boxed{
\boxed{\begin{array}{c} \mathsf{t} \\ \Vert\,{\scriptstyle \mathsf{i}\downarrow,\mathsf{s}} \\ (e \wedge f) \vee \bar{e} \vee \bar{f} \end{array}}
\wedge c \wedge
\boxed{\begin{array}{c} \mathsf{t} \\ \Vert\,{\scriptstyle \mathsf{i}\downarrow,\mathsf{s}} \\ (e \wedge f) \vee \bar{e} \vee \bar{f} \end{array}}
\wedge d
}
\end{array}}
\vee
\left(
\boxed{\begin{array}{c} \mathsf{t} \\ \Vert\,{\scriptstyle \mathsf{i}\downarrow,\mathsf{s}} \\ (e \wedge f) \vee \bar{e} \vee \bar{f} \end{array}}
\wedge [\bar{c} \vee \bar{d}] \wedge a \wedge
\boxed{\begin{array}{c} \mathsf{t} \\ \Vert\,{\scriptstyle \mathsf{i}\downarrow,\mathsf{s}} \\ (e \wedge f) \vee \bar{e} \vee \bar{f} \end{array}}
\wedge [\bar{c} \vee \bar{d}] \wedge b
\right)
} \\
\Big\Vert\,{\scriptstyle \mathsf{s}} \\
\boxed{
\boxed{\begin{array}{c}
(e \wedge f) \vee (e \wedge f) \vee (e \wedge f) \vee (e \wedge f) \\
\Vert\,{\scriptstyle \mathsf{c}\downarrow,\mathsf{m}} \\
e \wedge f
\end{array}}
\vee ([\bar{e} \vee \bar{f}] \wedge c \wedge [\bar{e} \vee \bar{f}] \wedge d) \vee ([\bar{e} \vee \bar{f}] \wedge [\bar{c} \vee \bar{d}] \wedge a \wedge [\bar{e} \vee \bar{f}] \wedge [\bar{c} \vee \bar{d}] \wedge b)
}
\end{array}
}
\vee \bar{a} \vee \bar{b}
\end{array}
$$

In open deduction analytic Statman proofs grow polynomially.

Deep inference and proof complexity (size)



$\longrightarrow$ = 'polynomially simulates'.

Open deduction:

- in the cut-free case, thanks to deep inference, has an exponential speed-up over the cut-free sequent calculus (e.g., over Statman tautologies)—1, see [6];
- has as small proofs as the best formalisms—2, 3, 4, 5, see [6];
- thanks to dagness, has quasipolynomial cut elimination (instead of exponential) [7, 15].
- Cut free deep inference outperforms the sequent calculus.

Deep inference and proof search complexity

Unconstrained bottom-up formula-driven proof search has horrendous complexity due to deep inference, because every connective can make the search tree branch.

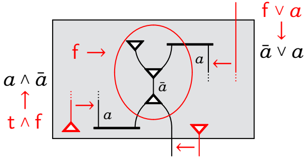
However:

1. Das proved that in the presence of distributivity, a depth 2 proof system polynomially simulates any unbounded depth proof system [8]. This means that a very moderate increase of nondeterminism buys exponentially smaller proofs.
2. Focusing techniques should be facilitated by the more liberal proof composition.
3. In particular it should be possible to confine the search inside small sub-spaces of canonical proofs.
4. The sequent calculus was designed to make proof search finite, not necessarily to make it efficient.

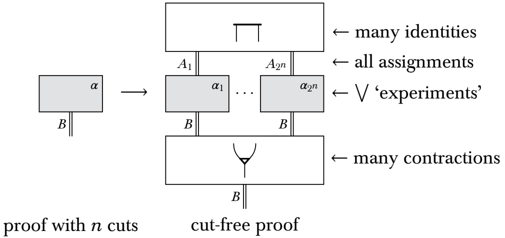
Cut elimination by ‘experiments’ (for logics with contraction)

Experiment

over a proof:



We do:



- Simple, exponential cut elimination;
- 2^n experiments, where n is the number of atoms;
- fairly syntax independent method.

The secret of success is in the proof composition mechanism.

Why is this impossible in the sequent calculus?

Normalisation in the linear fragment: Splitting

Theorem (Splitting) *For every proof* $\begin{array}{c}\mathsf{t}\\ \Vert\\ K\{A \wedge B\}\end{array}$ *there are proofs*

$$\begin{array}{c}K_A \vee K_B \vee \{\ \}\\ \Vert\\ K\{\ \}\end{array} \qquad \begin{array}{c}\mathsf{t}\\ \Vert\\ K_A \vee A\end{array} \qquad \begin{array}{c}\mathsf{t}\\ \Vert\\ K_B \vee B\end{array}$$

Similar theorems hold for every logics we tried so far (including logics that for Gentzen theory are hopeless).

Splitting for an atomic cut

Therefore for every cut-free proof $\begin{array}{c}\mathsf{t}\\ \Vert \\ K\{a \wedge \bar{a}\}\end{array}$ there are cut-free proofs

$$\begin{array}{c} K'\{\bar{a}\} \vee K''\{a\} \vee \{\ \} \\ \Vert \\ K\{\ \} \end{array} \qquad \begin{array}{c}\mathsf{t}\\ \Vert \\ K'\{\bar{a}\} \vee a\end{array} \qquad \begin{array}{c}\mathsf{t}\\ \Vert \\ K''\{a\} \vee \bar{a}\end{array}$$

therefore we can build $\begin{array}{c} \mathsf{i}{\downarrow}\dfrac{\mathsf{t}}{\boxed{\boxed{\begin{array}{c} a \\ \Vert \\ K'\{\bar{a}\}\end{array}} \vee \boxed{\begin{array}{c}\bar{a}\\ \Vert \\ K''\{a\}\end{array}}}} \\ \Vert \\ K\{\mathsf{f}\} \end{array}$

therefore a cut at the is admissible.

Ingredients for a Kleene algebra

Two things are necessary:

1. A sequentiality operator: $a.b$;
2. The Kleene star: $a^\star = \{\epsilon, a, a.a, a.a.a, \dots\}$.

This stuff is the basis of many process algebras, *e.g.*, CCS.

Surprise(?): sequentiality cannot be captured in an analytic Gentzen system. It requires deep inference [9, 22].

The moral reason is that sequentiality and the Kleene's star are self-dual and noncommutative:

$$a.b \mid \overline{a.b} = a.b \mid \bar{a}.\bar{b} \to \circ \quad .$$

Technically, Tiu's counterexample applies (see next slides and [22, 23]).

System BV

BV = MLL + self-dual noncommutative operator [9, 22]:

- Equations:

$$\overline{A \otimes B} = \bar{A} \parr \bar{B} \qquad \overline{A \parr B} = \bar{A} \otimes \bar{B} \qquad \overline{A \lhd B} = \bar{A} \lhd \bar{B}$$

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C$$
$$A \lhd \langle B \lhd C \rangle = \langle A \lhd B \rangle \lhd C$$
$$A \parr [B \parr C] = [A \parr B] \parr C$$

$$A \otimes B = B \otimes A \qquad A \parr B = B \parr A$$

$$A \otimes \circ = A \lhd \circ = \circ \lhd A = A \parr \circ = A$$

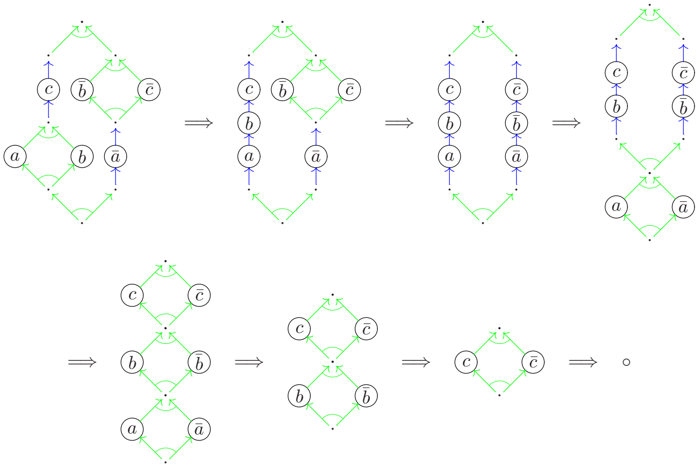
- Rules:

$$\mathsf{i}{\uparrow}\,\frac{a \otimes \bar{a}}{\circ} \qquad\qquad \mathsf{q}{\uparrow}\,\frac{\langle A \lhd B \rangle \otimes \langle C \lhd D \rangle}{(A \otimes C) \lhd (B \otimes D)}$$

$$\mathsf{i}{\downarrow}\,\frac{\circ}{a \parr \bar{a}} \qquad \mathsf{s}\,\frac{A \otimes [B \parr C]}{(A \otimes B) \parr C} \qquad \mathsf{q}{\downarrow}\,\frac{[A \parr C] \lhd [B \parr D]}{\langle A \lhd B \rangle \parr \langle C \lhd D \rangle}$$

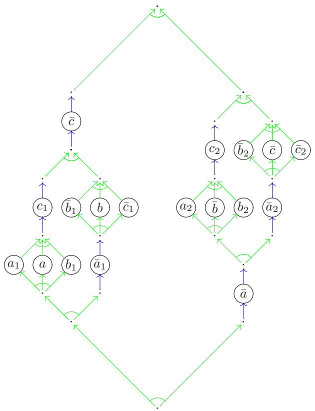
Tiu's counterexample: BV is not expressible in Gentzen

Graphical representation of a proof in BV:



Tiu's counterexample: BV is not expressible in Gentzen (cont.)

We can build a growing fractal of growing depth; the next step is:



…and each of its cut-free proofs has to start deeper inside.

Therefore BV cannot be captured by shallow inference!

Splitting for BV

Theorem (Splitting) *For every proof* $\begin{array}{c}\circ\\ \Vert\\ K\{A \otimes B\}\end{array}$ *there are proofs*

$$\begin{array}{c}K_A \mathbin{\bindnasrepma} K_B \mathbin{\bindnasrepma} \{\ \}\\ \Vert\\ K\{\ \}\end{array} \qquad \begin{array}{c}\circ\\ \Vert\\ K_A \mathbin{\bindnasrepma} A\end{array} \qquad \begin{array}{c}\circ\\ \Vert\\ K_B \mathbin{\bindnasrepma} B\end{array}$$

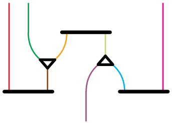
and for every proof $\begin{array}{c}\circ\\ \Vert\\ K\{A \triangleleft B\}\end{array}$ *there are proofs*

$$\begin{array}{c}\langle K_A \triangleleft K_B\rangle \mathbin{\bindnasrepma} \{\ \}\\ \Vert\\ K\{\ \}\end{array} \qquad \begin{array}{c}\circ\\ \Vert\\ K_A \mathbin{\bindnasrepma} A\end{array} \qquad \begin{array}{c}\circ\\ \Vert\\ K_B \mathbin{\bindnasrepma} B\end{array}$$

Splitting recovers Gentzen's notion of analyticity without imposing it on the meta-level of the formalism.

Atomic flows – Locality yields topology

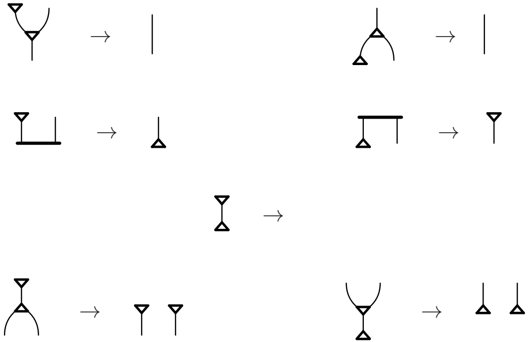
$$\mathsf{s}\,\dfrac{\mathsf{m}\,\dfrac{\dfrac{\mathsf{t}}{a \vee \bar{a}}}{[a \vee \mathsf{t}] \wedge [\mathsf{t} \vee \bar{a}]}}{\left[\mathsf{s}\,\dfrac{[a \vee \mathsf{t}] \wedge \bar{a}}{\dfrac{a \wedge \bar{a}}{\mathsf{f}} \vee \mathsf{t}} \vee \mathsf{t}\right]} \quad = \dfrac{\left(\mathsf{s}\,\dfrac{a \wedge \left[\bar{a} \vee \dfrac{\mathsf{t}}{\bar{a} \vee a}\right]}{\dfrac{a \wedge \dfrac{\bar{a} \vee \bar{a}}{\bar{a}}}{\mathsf{f}} \vee \dfrac{a}{a \wedge a}} \wedge \bar{a}\right)}{a \wedge \dfrac{a \wedge \bar{a}}{\mathsf{f}}} \qquad \mathsf{m}\,\dfrac{\dfrac{a}{a \wedge a} \vee \dfrac{b}{b \wedge b}}{[a \vee b] \wedge [a \vee b]} \wedge \dfrac{a}{a \wedge a}$$



Below the proofs, their (atomic) flows [10] are shown:

- only structural information is retained in flows;
- logical information is lost;
- flow size is polynomially related to derivation size;
- composition of proofs naturally correspond to composition of flows.

Flow reductions: (co)weakening (I)



Each flow reduction corresponds to a correct proof reduction.

Flow reductions: (co)weakening (2)

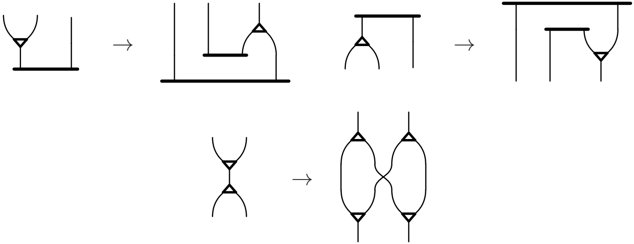
E.g.,  $\rightarrow$  specifies that

$$
\begin{array}{c}
\Pi'' \Big\Vert \\
\xi\left\{\dfrac{\mathsf{t}}{a^{\epsilon} \vee \bar{a}}\right\} \\
\Phi \Big\Vert \\
\zeta\left\{\dfrac{a^{\epsilon}}{\mathsf{t}}\right\} \\
\Psi \Big\Vert \\
\alpha
\end{array}
\qquad \text{becomes} \qquad
\begin{array}{c}
\Pi'' \Big\Vert \\
\xi\left[\mathsf{t} \vee \dfrac{\mathsf{f}}{\bar{a}}\right] \\
\Phi\{a^{\epsilon}/\mathsf{t}\} \Big\Vert \\
\zeta\{\mathsf{t}\} \\
\Psi \Big\Vert \\
\alpha
\end{array}
$$

We can operate on flow reductions instead than on derivations:

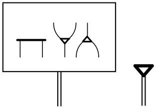
- much easier,
- we get natural, syntax-independent induction measures.

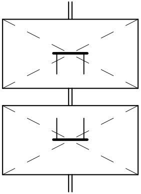
Flow reductions: (co)contraction



- These reductions conserve the number and length of paths.
- Open problem: does cocontraction yield superpolynomial compression?

Generalising the cut-free form

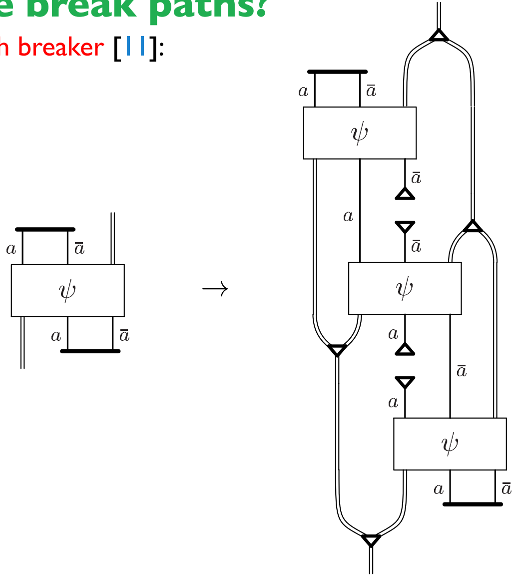
- Normalised proof: 

- Normalised derivation: 

- The symmetric form is called streamlined.
- Cut elimination is a corollary of streamlining.
- We just need to break the paths between identities and cuts, and (co)weakenings do the rest.

How do we break paths?

With the path breaker [11]:



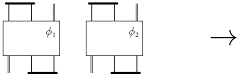
Even if there is a path between identity and cut on the left, there is none on the right.

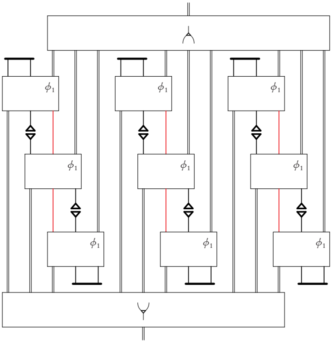
We can do the same on derivations, of course

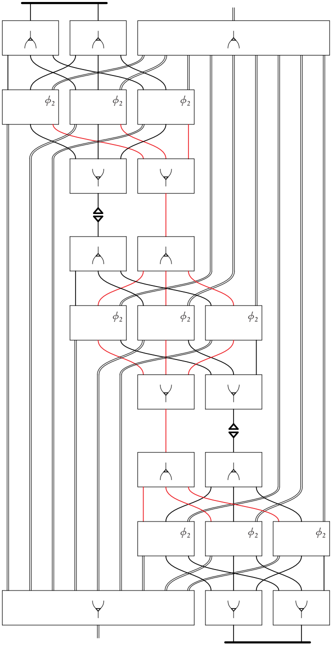
$$\dfrac{\dfrac{A}{[a \vee \bar{a}] \wedge A} \atop {\Psi \Vert}}{\dfrac{B \vee (a \wedge \bar{a})}{B}} \quad \longrightarrow \quad \begin{array}{c} A \\ \Vert \{\mathsf{c}\uparrow, \mathsf{ai}\downarrow, =\} \\ (([a \vee \bar{a}] \wedge A) \wedge A) \wedge A \\ {\scriptstyle (\Psi \wedge A) \wedge A} \Vert \\ ([B \vee (a \wedge \bar{a})] \wedge A) \wedge A \\ {\scriptstyle \Phi_a \wedge A} \Vert \\ [B \vee ([a \vee \bar{a}] \wedge A)] \wedge A \\ {\scriptstyle [B \vee \Psi] \wedge A} \Vert \\ B \vee ([B \vee (a \wedge \bar{a})] \wedge A) \\ {\scriptstyle B \vee \Phi_{\bar{a}}} \Vert \\ B \vee [B \vee ([a \vee \bar{a}] \wedge A)] \\ {\scriptstyle B \vee [B \vee \Psi]} \Vert \\ B \vee [B \vee [B \vee (a \wedge \bar{a})]] \\ \Vert \{\mathsf{c}\downarrow, \mathsf{ai}\uparrow, =\} \\ B \end{array}$$

- We can compose this as many times as there are paths between identities and cut.
- We obtain a family of normalisers that only depends on *n*.
- The construction is exponential.
- Finding something like this is unthinkable without flows.

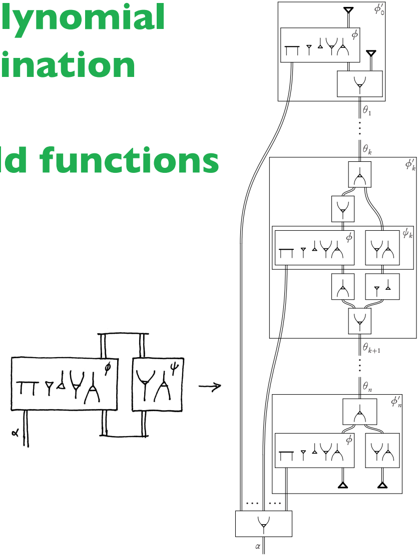
Example for 2 cuts







Quasipolynomial cut elimination by threshold functions



- Only $n + 1$ copies of the proof are stitched together.
- Note local cocontraction (= better sharing, not available in Gentzen).

Conclusions

- Composition in Gentzen is too rigid (it was designed for consistency proofs, not much else).
- Deep inference composition is free and yields local proof systems.
- Locality = linearity + atomicity, so we are doing an extreme form of linear logic.
- Because of locality we obtain a sort of geometric control over proofs.
- So we obtain an efficient and natural formalism for proofs, where more proof theory can be done with lower complexity.
- We are obtaining interesting notions of proof semantics.

References

[1] Richard F. Blute, Alessio Guglielmi, Ivan T. Ivanov, Prakash Panangaden & Lutz Straßburger (2014): *A Logical Basis for Quantum Evolution and Entanglement.*
In Claudia Casadio, Bob Coecke, Michael Moortgat & Philip Scott, editors: *Categories and Types in Logic, Language, and Physics*, *Lecture Notes in Computer Science* 8222, Springer-Verlag, pp. 90–107, doi:10.1007/978-3-642-54789-8_6.
Available at `http://cs.bath.ac.uk/ag/p/LBQEE.pdf`.

[2] Kai Brünnler (2004): *Deep Inference and Symmetry in Classical Proofs.*
Logos Verlag, Berlin.
Available at `http://cs.bath.ac.uk/ag/kai/phd.pdf`.

[3] Kai Brünnler (2010): *Nested Sequents.*
Available at `http://arxiv.org/pdf/1004.1845`.
Habilitation Thesis.

[4] Kai Brünnler & Alwen Fernanto Tiu (2001): *A Local System for Classical Logic.*
In R. Nieuwenhuis & Andrei Voronkov, editors: *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, *Lecture Notes in Computer Science* 2250, Springer-Verlag, pp. 347–361, doi:10.1007/3-540-45653-8_24.
Available at `http://cs.bath.ac.uk/ag/kai/lcl-lpar.pdf`.

[5] Paola Bruscoli (2002): *A Purely Logical Account of Sequentiality in Proof Search.*
In Peter J. Stuckey, editor: *Logic Programming, 18th International Conference (ICLP)*, *Lecture Notes in Computer Science* 2401, Springer-Verlag, pp. 302–316, doi:10.1007/3-540-45619-8_21.
Available at `http://cs.bath.ac.uk/pb/bvl/bvl.pdf`.

References (cont.)

[6] Paola Bruscoli & Alessio Guglielmi (2009): *On the Proof Complexity of Deep Inference.* *ACM Transactions on Computational Logic* 10(2), pp. 14:1–34, doi:10.1145/1462179.1462186. Available at http://cs.bath.ac.uk/ag/p/PrComplDI.pdf.

[7] Paola Bruscoli, Alessio Guglielmi, Tom Gundersen & Michel Parigot (2010): *A Quasipolynomial Cut-Elimination Procedure in Deep Inference Via Atomic Flows and Threshold Formulae.* In Edmund M. Clarke & Andrei Voronkov, editors: *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-16)*, *Lecture Notes in Computer Science* 6355, Springer-Verlag, pp. 136–153, doi:10.1007/978-3-642-17511-4_9. Available at http://cs.bath.ac.uk/ag/p/QPNDI.pdf.

[8] Anupam Das (2011): *On the Proof Complexity of Cut-Free Bounded Deep Inference.* In Kai Brünnler & George Metcalfe, editors: *Tableaux 2011*, *Lecture Notes in Artificial Intelligence* 6793, Springer-Verlag, pp. 134–148, doi:10.1007/978-3-642-22119-4_12. Available at http://www.anupamdas.com/items/PrCompII/ProofComplexityBoundedDI.pdf.

[9] Alessio Guglielmi (2007): *A System of Interaction and Structure.* *ACM Transactions on Computational Logic* 8(1), pp. 1:1–64, doi:10.1145/1182613.1182614. Available at http://cs.bath.ac.uk/ag/p/SystIntStr.pdf.

[10] Alessio Guglielmi & Tom Gundersen (2008): *Normalisation Control in Deep Inference Via Atomic Flows.* *Logical Methods in Computer Science* 4(1), pp. 9:1–36, doi:10.2168/LMCS-4(1:9)2008. Available at http://arxiv.org/pdf/0709.1205.pdf.

References (cont.)

[11] Alessio Guglielmi, Tom Gundersen & Lutz Straßburger (2010): *Breaking Paths in Atomic Flows for Classical Logic.*
In Jean-Pierre Jouannaud, editor: *25th Annual IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, pp. 284–293, doi:10.1109/LICS.2010.12.
Available at http://www.lix.polytechnique.fr/~lutz/papers/AFII.pdf.

[12] Tom Gundersen (2009): *A General View of Normalisation Through Atomic Flows.*
Ph.D. thesis, University of Bath.
Available at
http://tel.archives-ouvertes.fr/docs/00/50/92/41/PDF/thesis.pdf.

[13] Tom Gundersen, Willem Heijltjes & Michel Parigot (2013): *Atomic Lambda Calculus: A Typed Lambda-Calculus with Explicit Sharing.*
In Orna Kupferman, editor: *28th Annual IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, pp. 311–320, doi:10.1109/LICS.2013.37.
Available at http://opus.bath.ac.uk/34527/1/AL.pdf.

References (cont.)

[14] Tom Gundersen, Willem Heijltjes & Michel Parigot (2013): *A Proof of Strong Normalisation of the Typed Atomic Lambda-Calculus*.
In Ken McMillan, Aart Middeldorp & Andrei Voronkov, editors: *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-19)*, *Lecture Notes in Computer Science* 8312, Springer-Verlag, pp. 340–354, doi:10.1007/978-3-642-45221-5_24.
Available at http://www.cs.bath.ac.uk/~wbh22/pdf/strong-normalisation-atomic-lambda-gundersen-heijltjes-parigot-2013.pdf.

[15] Emil Jeřábek (2009): *Proof Complexity of the Cut-Free Calculus of Structures*.
Journal of Logic and Computation 19(2), pp. 323–339, doi:10.1093/logcom/exn054.
Available at http://www.math.cas.cz/~jerabek/papers/cos.pdf.

[16] Ozan Kahramanoğulları (2005): *Towards Planning as Concurrency*.
In M.H. Hamza, editor: *Artificial Intelligence and Applications (AIA)*, ACTA Press, pp. 197–202.
Available at http://www.wv.inf.tu-dresden.de/~guglielm/ok/aia05.pdf.

[17] Luca Roversi (2011): *Linear Lambda Calculus and Deep Inference*.
In Luke Ong, editor: *Typed Lambda Calculi and Applications*, *Lecture Notes in Computer Science* 6690, Springer-Verlag, pp. 184–197, doi:10.1007/978-3-642-21691-6_16.
Available at http://www.di.unito.it/~rover/RESEARCH/PUBLICATIONS/2011-TLCA/Roversi2011TLCA.pdf.

References (cont.)

[18] Luca Roversi (2014): *A Deep Inference System with a Self-Dual Binder Which Is Complete for Linear Lambda Calculus*.
Journal of Logic and Computation, doi:10.1093/logcom/exu033.
Available at http://www.di.unito.it/~rover/RESEARCH/PUBLICATIONS/2014-JLC/Roversi2014JLC.pdf.
To appear.

[19] Lutz Straßburger (2003): *Linear Logic and Noncommutativity in the Calculus of Structures*.
Ph.D. thesis, Technische Universität Dresden.
Available at
http://www.lix.polytechnique.fr/~lutz/papers/dissvonlutz.pdf.

[20] Lutz Straßburger (2006): *Proof Nets and the Identity of Proofs*.
Technical Report 6013, INRIA.
Available at http://hal.inria.fr/docs/00/11/43/20/PDF/RR-6013.pdf.

[21] Lutz Straßburger (2011): *From Deep Inference to Proof Nets Via Cut Elimination*.
Journal of Logic and Computation 21(4), pp. 589–624.
Available at http://www.lix.polytechnique.fr/~lutz/papers/deepnet.pdf.

[22] Alwen Tiu (2006): *A System of Interaction and Structure II: The Need for Deep Inference*.
Logical Methods in Computer Science 2(2), pp. 4:1–24, doi:10.2168/LMCS-2(2:4)2006.
Available at http://arxiv.org/pdf/cs.LO/0512036.pdf.

References (cont.)

[23] Alwen Fernanto Tiu (2001): *Properties of a Logical System in the Calculus of Structures*. Technical Report WV-01-06, Technische Universität Dresden. Available at http://users.cecs.anu.edu.au/~tiu/thesisc.pdf.