Normalisation with Atomic Flows

Alessio Guglielmi

University of Bath and LORIA & INRIA Nancy-Grand Est

Joint work with Paola Bruscoli, Tom Gundersen and Michel Parigot
January 2010
This talk is available at http://cs.bath.ac.uk/ag/t/NAF.pdf

Outline

(Proof) System SKS [Brünnler & Tiu(2001)]

- Atomic rules:

$\mathsf{ai}\!\downarrow \dfrac{\mathsf{t}}{a \vee \bar{a}}$	$\mathsf{aw}\!\downarrow \dfrac{\mathsf{f}}{a}$	$\mathsf{ac}\!\downarrow \dfrac{a \vee a}{a}$
identity	*weakening*	*contraction*
$\mathsf{ai}\!\uparrow \dfrac{a \wedge \bar{a}}{\mathsf{f}}$	$\mathsf{aw}\!\uparrow \dfrac{a}{\mathsf{t}}$	$\mathsf{ac}\!\uparrow \dfrac{a}{a \wedge a}$
cut	*coweakening*	*cocontraction*

- Linear rules:

$\mathsf{s}\,\dfrac{A \wedge [B \vee C]}{(A \wedge B) \vee C}$	$\mathsf{m}\,\dfrac{(A \wedge B) \vee (C \wedge D)}{[A \vee C] \wedge [B \vee D]}$
switch	*medial*

- Plus an '$=$' linear rule (associativity, commutativity, units).
- Rules are applied anywhere inside formulae.
- Negation on atoms only.
- Cut is atomic.
- SKS is complete and implicationally complete for propositional logic.

Example 1

- In the calculus of structures (CoS):

$$
\begin{array}{rc}
 & [a \vee b] \wedge a \\ \cline{2-2}
\mathsf{ac}\!\uparrow & [(a \wedge a) \vee b] \wedge a \\ \cline{2-2}
\mathsf{ac}\!\uparrow & [(a \wedge a) \vee (b \wedge b)] \wedge a \\ \cline{2-2}
\mathsf{ac}\!\uparrow & [(a \wedge a) \vee (b \wedge b)] \wedge (a \wedge a) \\ \cline{2-2}
\mathsf{m} & ([a \vee b] \wedge [a \vee b]) \wedge (a \wedge a) \\ \cline{2-2}
= & ([a \vee b] \wedge a) \wedge ([a \vee b] \wedge a)
\end{array}
$$

- In 'Formalism A':

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

Top-down symmetry: so inference steps can be made atomic (the medial rule, m, is impossible in the sequent calculus).

Example 2

- In CoS:

$$
\begin{array}{rc}
 & \mathsf{t} \\
\mathsf{ai}\!\downarrow & \overline{a \vee \bar{a}} \\
= & \overline{(a \wedge \mathsf{t}) \vee (\mathsf{t} \wedge \bar{a})} \\
\mathsf{m} & \overline{[a \vee \mathsf{t}] \wedge [\mathsf{t} \vee \bar{a}]} \\
= & \overline{[a \vee \mathsf{t}] \wedge [\bar{a} \vee \mathsf{t}]} \\
\mathsf{s} & \overline{([a \vee \mathsf{t}] \wedge \bar{a}) \vee \mathsf{t}} \\
= & \overline{(\bar{a} \wedge [a \vee \mathsf{t}]) \vee \mathsf{t}} \\
\mathsf{s} & \overline{[(\bar{a} \wedge a) \vee \mathsf{t}] \vee \mathsf{t}} \\
= & \overline{(a \wedge \bar{a}) \vee \mathsf{t}} \\
\mathsf{ai}\!\uparrow & \overline{\mathsf{f} \vee \mathsf{t}} \\
= & \overline{\mathsf{t}}
\end{array}
$$

- In ‘Formalism A’:

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

Locality

- Deep inference allows locality,
- *i.e.*, inference steps can be checked in constant time (so, inference steps are small).

Example, atomic cocontraction:

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

Note: the sequent calculus

- does not allow locality in contraction (counterexample in [Brünnler(2004)]), and
- does not allow local reduction of cut into atomic form.

Goal of This Talk

To illustrate the slogans:

- Deep inference = locality (+ symmetry).
- Locality = linearity + atomicity.
- Geometry = syntax independence (elimination of bureaucracy).
- Locality → geometry → semantics of proofs (Lamarche *dixit*).

This is a path towards solving the problem of proof identity, *i.e.*, determining when two proofs are the same (Hilbert's '24th problem').

To show that:

- We can normalise in a somewhat syntax-independent way.
- Normalisation is a very robust phenomenon.
- Perhaps traditional proof theory is prejudiced on analyticity and complexity: analyticity is much cheaper than exponential!

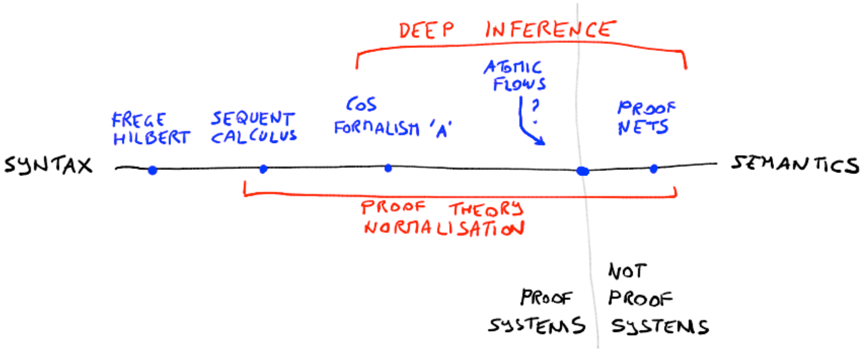
What Do We Need to Solve the Proof Identity Problem?

A finer representation of proofs, achieving locality.

This yields:

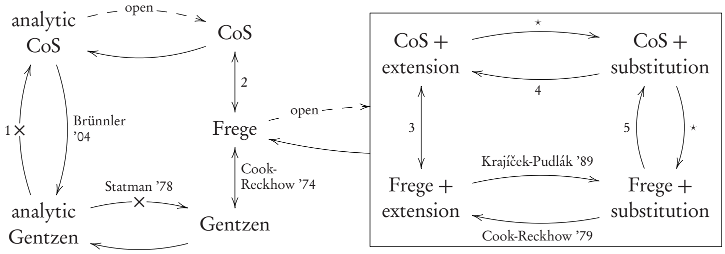
- more proofs to choose representatives from, and especially
- bureaucracy-free proofs;
- nice geometric models [Guiraud(2006)];
- smaller proofs, but
- not as small as proof nets [Lamarche & Straßburger(2005)];
- more manipulation possibilities, *viz.*, for normalisation (focus of this talk, and where we got surprises).

Elimination of Bureaucracy



- Propositional logic.
- Proof system ≈ proofs can be checked in polytime.
- Normalisation = mainly, but not only!, cut elimination.
- Objective: eliminate bureaucracy, *i.e.*, find 'something' at the boundary.

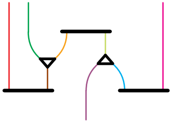
What About Proof Complexity?



Deep inference has as small proofs as the best proof systems
and
it has a normalisation theory
and
its analytic proof systems are more powerful than Gentzen ones
and
cut elimination is quasipolynomial (instead of exponential).
(See [Jeřábek(2009), Bruscoli & Guglielmi(2009), Bruscoli et al.(2009)Bruscoli, Guglielmi, Gundersen, & Parigot]).

(Atomic) Flows

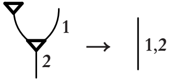
$$\mathsf{s}\,\dfrac{\mathsf{m}\,\dfrac{\dfrac{\mathsf{t}}{a\vee\bar a}}{[a\vee\mathsf{t}]\wedge[\mathsf{t}\vee\bar a]}}{\left[\mathsf{s}\,\dfrac{[a\vee\mathsf{t}]\wedge\bar a}{\dfrac{a\wedge\bar a}{\mathsf{f}}\vee\mathsf{t}}\quad\vee\quad\mathsf{t}\right]}
= \dfrac{\left(\mathsf{s}\,\dfrac{a\wedge\left[\bar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

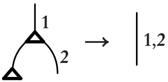


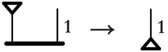
- Below derivations, their (atomic) flows are shown.
- Only structural information is retained in flows.
- Logical information is lost.
- Flow size is polynomially related to derivation size.

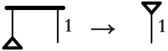
Flow Reductions: (Co)Weakening (1)

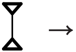
Consider these flow reductions:

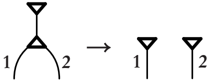
aw↓-ac↓: 

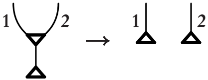
ac↑-aw↑: 

aw↓-ai↑: 

ai↓-aw↑: 

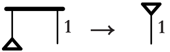
aw↓-aw↑: 

aw↓-ac↑: 

ac↓-aw↑: 

Each of them corresponds to a correct derivation reduction.

Flow Reductions: (Co)Weakening (2)

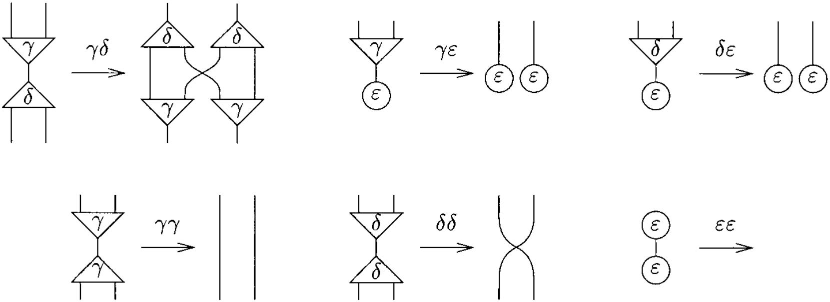
For example, ai↓-aw↑:  specifies that

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

We can operate on flow reductions instead than on derivations: it is much easier and we get natural, syntax-independent induction measures.

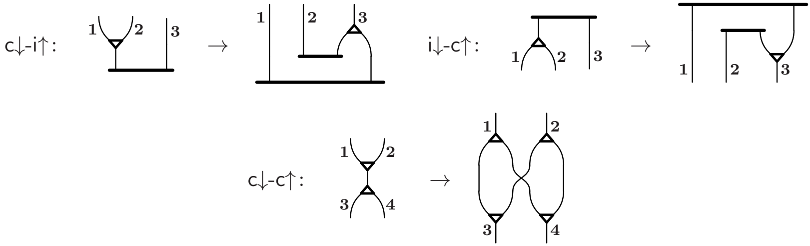
Relation With Interaction Combinators?

Lots of coincidences, but also differences: no apparent logical meaning for two ‘contractions’:



Flow Reductions: (Co)Contraction

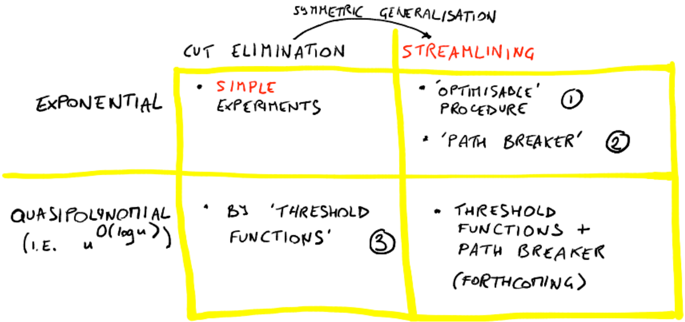
Consider these flow reductions:



- They conserve the number and length of paths.
- Note that they can blow up a derivation exponentially.
- It's a good thing: cocontraction is a new compression mechanism (sharing?).
- Open problem: does cocontraction provide exponential compression? Conjecture: yes.

Normalisation

Overview

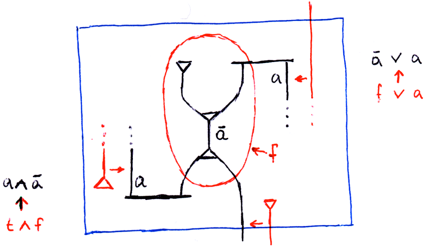


- None of these methods existed before atomic flows, none of them requires permutations or other syntactic devices.
- Quasipolynomial procedures are surprising.

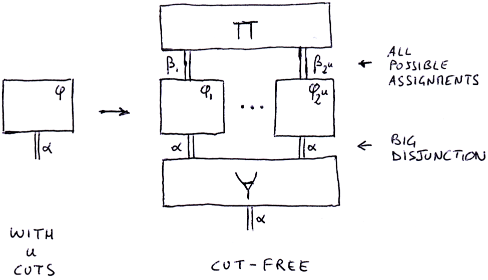
(1) [Guglielmi & Gundersen(2008)]; (2) LICS 2010 submission; (3) [Bruscoli et al.(2009)Bruscoli, Guglielmi, Gundersen, & Parigot].

Cut Elimination (on Proofs) by 'Experiments'

Experiment:



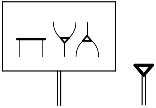
We do:



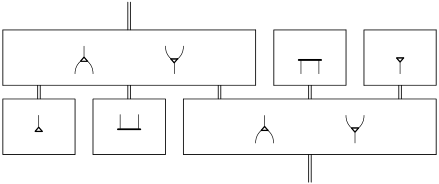
Simple, exponential cut elimination; proof generates 2^n experiments.

Generalising the Cut-Free Form

- Normalised proof:



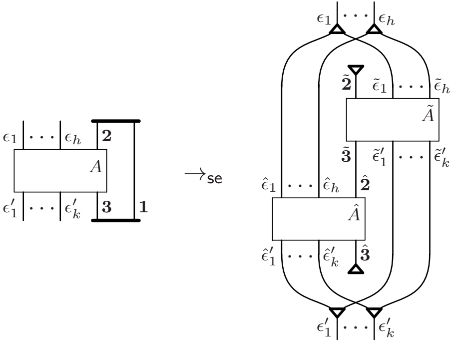
- Normalised derivation:



- The symmetric form is called streamlined.
- Cut elimination is a corollary of streamlining.

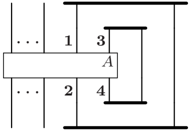
Removal of a 'Simple Edge'

Remove identity and cut:

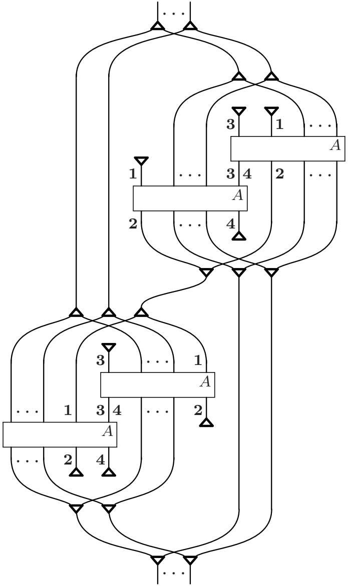


- We can do so on simple edges, like **1** above.
- The procedure requires a strategy, not to loop.
- The chunks to be copied can be small.
- Open: computational interpretation?

Composition of Simple Edge Removal

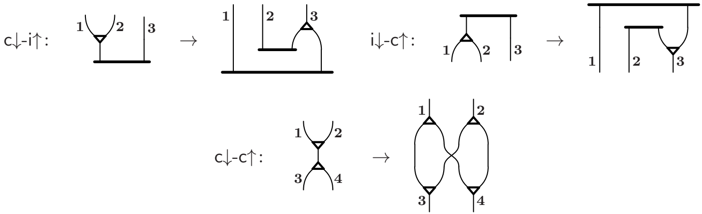


$\rightarrow_{\text{bc}}$

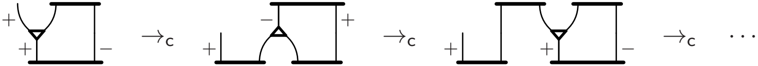


How to Obtain a Simple Edge?

- By moving away (co)contractions by way of their reductions:



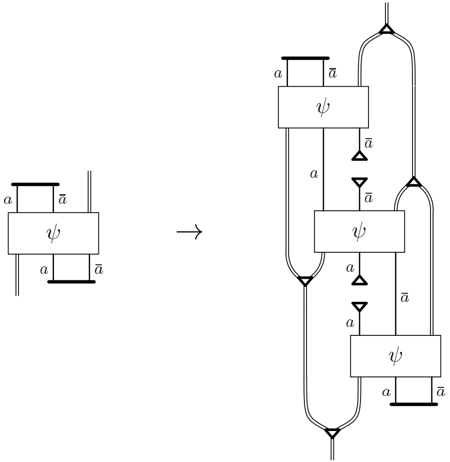
- But beware of loops:



- This and more is in [Guglielmi & Gundersen(2008)].

How Do We Break Paths Without 'Preprocessing'?

With the path breaker (Lutz Straßburger contributed here):



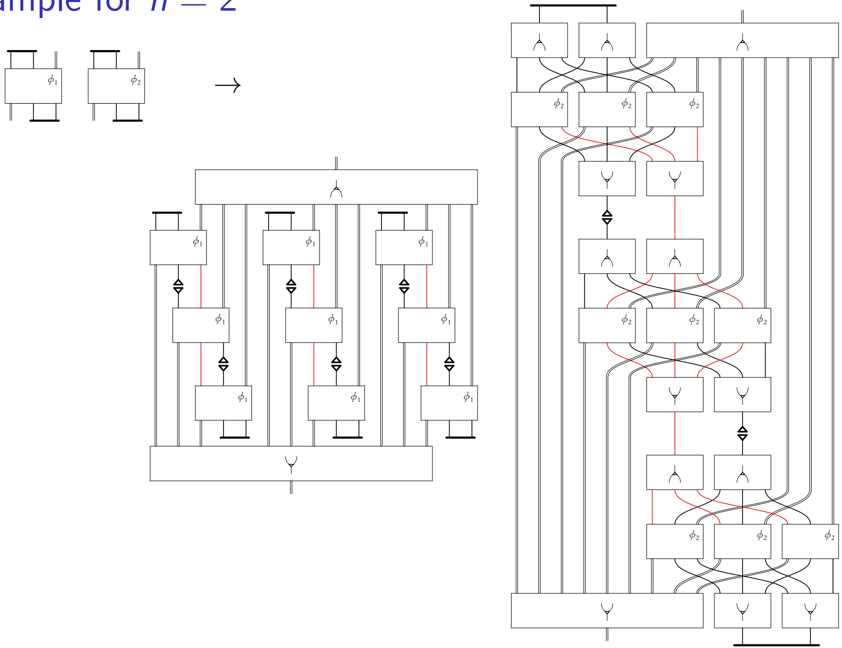
Even if there is a path between identity and cut on the left, there is none on the right.

We Can Do This on Derivations, of Course

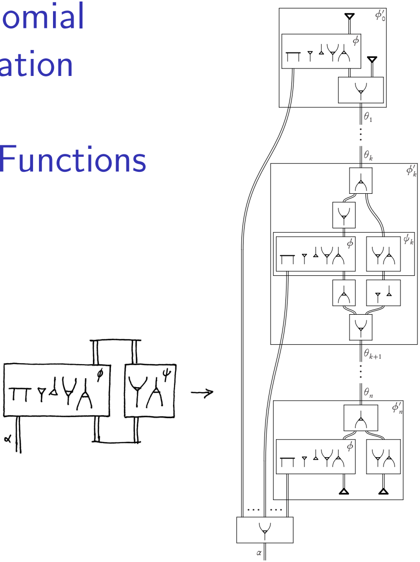
$$
\begin{array}{c}
A \\ \hline
[a \vee \bar{a}] \wedge A \\
\Psi \Big\| \\
B \vee (a \wedge \bar{a}) \\ \hline
B
\end{array}
\quad \longrightarrow \quad
\begin{array}{c}
A \\
\Big\| \{\mathsf{c}\uparrow, \mathsf{ai}\downarrow, =\} \\
(([a \vee \bar{a}] \wedge A) \wedge A) \wedge A \\
(\Psi \wedge A)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

- 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.
- Note: finding something like this is *unthinkable* without flows.

Example for $n = 2$



Quasipolynomial Cut Elimination by Threshold Functions



Only $n + 1$ copies of the proof are stitched together. It's complicated, but note local cocontraction ($=$ better sharing, not available in Gentzen).

Handwaving Explanation of Threshold Functions

- θ_i = there are at least i atoms that are true (out of given n).
- For example, for $n = 2$, we have $\theta_1 = a \vee b$ and $\theta_2 = a \wedge b$.
- Each θ_i can be kind of projected into each atom to provide its pseudocomplement, for example the pseudocomplement of a in θ_1 is b.
- The atom and the pseudocomplement fit into the scheme of the previous slide, and you can get, for example, θ_2 from θ_1.
- Stitch derivations together until you get $\theta_{n+1} = \mathsf{f}$.
- The complexity is dominated by the complexity of the θ's, which is $n^{O(\log n)}$.

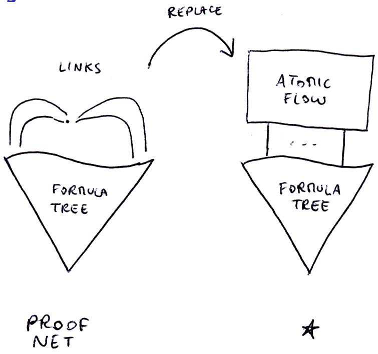
The difficulty is in defining the θ's and in finding proofs that stitch them together (this theory comes from circuit complexity and it had been applied to the monotone sequent calculus, which is weaker than propositional logic).

Conjecture 1

We can normalise in polynomial time, because:

- polynomial threshold function representations exist;
- deep inference is flexible.

Conjecture 2



- We think that (*) might make for a proof system (see also recent work by Straßburger).
- This means that there should exist a polynomial algorithm to check the correctness of (*).
- If this is true, we have an excellent bureaucracy-free formalism.
- Note: if such a thing existed for proof nets, then coNP = NP.

Conclusion

- Normalisation does not depend on logical rules.
- It only depends on structural information, *i.e.*, geometry.
- Normalisation is extremely robust.
- Deep inference's locality is key.
- Complexity-wise, deep inference is as powerful as the best formalisms,
- and more powerful if analiticity is requested.
- Deep inference is the continuation of Girard politics with other means.

In my opinion, much of the future of structural proof theory is in 'geometric methods'.

This talk is available at `http://cs.bath.ac.uk/ag/t/NAF.pdf`

References

Brünnler, K. (2004).
Deep Inference and Symmetry in Classical Proofs.
Berlin: Logos Verlag.
http://www.iam.unibe.ch/~kai/Papers/phd.pdf.

Brünnler, K., & Tiu, A. F. (2001).
A local system for classical logic.
In R. Nieuwenhuis, & A. Voronkov (Eds.) *LPAR 2001*, vol. 2250 of *Lecture Notes in Computer Science*, (pp. 347–361).
Springer-Verlag.
http://www.iam.unibe.ch/~kai/Papers/lcl-lpar.pdf.

Bruscoli, P., & Guglielmi, A. (2009).
On the proof complexity of deep inference.
ACM Transactions on Computational Logic, *10*(2), 1–34.
Article 14. http://cs.bath.ac.uk/ag/p/PrComplDI.pdf.

Bruscoli, P., Guglielmi, A., Gundersen, T., & Parigot, M. (2009).
Quasipolynomial normalisation in deep inference via atomic flows and threshold formulae.
Submitted. http://cs.bath.ac.uk/ag/p/QuasiPolNormDI.pdf.

Guglielmi, A., & Gundersen, T. (2008).
Normalisation control in deep inference via atomic flows.
Logical Methods in Computer Science, *4*(1:9), 1–36.
http://www.lmcs-online.org/ojs/viewarticle.php?id=341.

Guiraud, Y. (2006).
The three dimensions of proofs.
Annals of Pure and Applied Logic, *141*(1-2), 266–295.
http://www.loria.fr/~guiraudy/recherche/cos1.pdf.

Jeřábek, E. (2009).
Proof complexity of the cut-free calculus of structures.
Journal of Logic and Computation, *19*(2), 323–339.
http://www.math.cas.cz/~jerabek/papers/cos.pdf.

Lamarche, F., & Straßburger, L. (2005).
Naming proofs in classical propositional logic.
In P. Urzyczyn (Ed.) *Typed Lambda Calculi and Applications*, vol. 3461 of *Lecture Notes in Computer Science*, (pp. 246–261).
Springer-Verlag.
http://www.lix.polytechnique.fr/~lutz/papers/namingproofsCL.pdf.