

# Ten Years of Deep Inference

Alessio Guglielmi

University of Bath and LORIA & INRIA Nancy-Grand Est

Joint work with Paola Bruscoli, Tom Gundersen and Michel Parigot  
(plus others that I'll mention)

January 2010

This talk is available at <http://cs.bath.ac.uk/ag/t/TYDI.pdf>

# Outline

Overview of Complexity Classes

Proof Systems

Compressing Proofs

Deep Inference

Proof Complexity and Deep Inference

Propositional Logic and System SKS

Goal of This Talk

Atomic Flows

Examples of Atomic Flows

Flow Reductions

Normalisation

Cut Elimination: Experiments

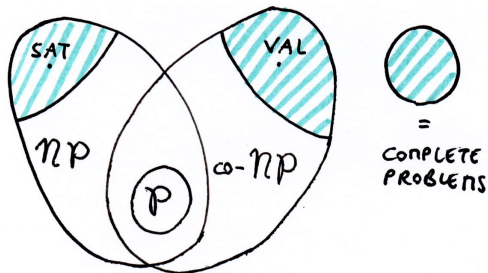
Generalised Cut Elimination and the Path Breaker

Quasipolynomial Cut Elimination

Elimination of Bureaucracy (a Problem of Hilbert!)

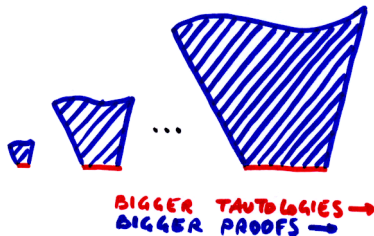
Conclusion

# Overview of (Some!) Complexity Classes



- ▶  $\mathcal{NP}$  = class of problems that are verifiable in polynomial time.
- ▶ SAT = 'Is a propositional formula satisfiable?' (Yes: here is a satisfying assignment.)
- ▶  $\text{co-}\mathcal{NP}$  = class of problems that are disqualifiable in polynomial time.
- ▶ VAL = 'Is a propositional formula valid?' (No: here is a falsifying assignment.)
- ▶  $\mathcal{P}$  = class of problems that can be solved in polynomial time.
- ▶  $\mathcal{NP} \neq \text{co-}\mathcal{NP}$  implies  $\mathcal{P} \neq \mathcal{NP}$ .

# Proof Systems



- ▶ Proof complexity = proof size.
- ▶ Proof system = algorithm that verifies proofs in polynomial time on their size.
- ▶ Important question: **What is the relation between size of tautologies and size of minimal proofs?**

# Example of Proof System: Frege

$$A \supset (B \supset A),$$

$$\text{Axioms: } (A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C)),$$

$$(\neg B \supset \neg A) \supset ((\neg B \supset A) \supset B),$$

$$\text{Modus ponens, or cut, rule: } \frac{A \quad A \supset B}{B}.$$

Example:

$$\frac{\frac{a \supset (a \supset a)}{a \supset (a \supset a)} \quad \frac{(a \supset ((a \supset a) \supset a)) \supset ((a \supset (a \supset a)) \supset (a \supset a))}{(a \supset ((a \supset a) \supset a)) \supset ((a \supset (a \supset a)) \supset (a \supset a))}}{\frac{(a \supset (a \supset a)) \supset ((a \supset ((a \supset a) \supset a)) \supset ((a \supset (a \supset a)) \supset (a \supset a)))}{(a \supset (a \supset a)) \supset ((a \supset ((a \supset a) \supset a)) \supset ((a \supset (a \supset a)) \supset (a \supset a)))}}{a \supset a}$$

Robustness: all Frege systems are polynomially equivalent.

## Example of Proof System: Gentzen Sequent Calculus

One axiom, many rules.

Example:

$$\begin{array}{c}
 \text{V}_{\text{RL}} \frac{a \vdash a}{a \vdash a \vee (a \supset \perp)} \quad a, \perp \vdash \perp \\
 \text{D}_{\text{L}} \frac{a, (a \vee (a \supset \perp)) \supset \perp \vdash \perp}{a, (a \vee (a \supset \perp)) \supset \perp \vdash \perp} \\
 \text{V}_{\text{L}} \frac{a \vee (a \supset \perp), (a \vee (a \supset \perp)) \supset \perp \vdash \perp}{a \vee (a \supset \perp) \vdash ((a \vee (a \supset \perp)) \supset \perp) \supset \perp} \\
 \text{D}_{\text{L}} \frac{a \vdash a \quad \perp, a \vdash \perp}{a \supset \perp, a \vdash \perp} \\
 \text{D}_{\text{R}} \frac{a \supset \perp, a \vdash \perp}{a \supset \perp \vdash a \supset \perp} \\
 \text{V}_{\text{RR}} \frac{a \supset \perp \vdash a \vee (a \supset \perp) \quad a \supset \perp, \perp \vdash \perp}{a \supset \perp \vdash (a \vee (a \supset \perp)) \supset \perp \vdash \perp} \\
 \text{D}_{\text{L}} \frac{a \supset \perp, (a \vee (a \supset \perp)) \supset \perp \vdash \perp}{a \supset \perp, (a \vee (a \supset \perp)) \supset \perp \vdash \perp}
 \end{array}$$

This is a special case of Frege, important because it admits complete and **analytic** proof systems (*i.e.*, cut-free proof systems, by which consistency proofs and proof-search algorithms can be obtained).

Frege and Gentzen systems are polynomially equivalent.

## Example of Proof System: Deep Inference

Proofs can be composed by the same operators as formulae.

Example:

$$= \frac{\left( \frac{\frac{a \wedge \frac{\bar{a} \vee \bar{a}}{f}}{a \wedge \frac{\bar{a}}{f}}}{s} \vee \frac{a}{a \wedge a} \right) \wedge \bar{a}}{a \wedge \frac{a \wedge \bar{a}}{f}}$$

This is a generalisation of Frege, which admits complete and **local** proof systems (*i.e.*, where steps can be verified in constant time).

Frege and deep-inference systems are polynomially equivalent.

The **calculus of structures** (CoS) is now a completely developed deep inference formalism.

# Proof Complexity and the $\mathcal{NP}$ Vs. $\text{co-}\mathcal{NP}$ Problem

- ▶ Theorem [Cook & Reckhow(1974)]:

*There exists an efficient proof system  
iff  
 $\mathcal{NP} = \text{co-}\mathcal{NP}$*

where 'efficient' = admitting proofs that are verifiable in polynomial time over the size of the proved formula.

- ▶ **Is there an always efficient proof system?** Probably not, and this is, obviously, **hard**.
- ▶ **Is there an optimal proof system?** (in the sense that it polynomially simulates all others.) We don't know, and this is **perhaps feasible**.



# Compressing Proofs 1

Thus, an important question is:

How can we make proofs smaller?

These are known mechanisms:

1. Use **higher orders** (for example, second order propositional, for propositional formulae).
2. Add **substitution**:  $\text{sub} \frac{A}{A\sigma}$ .
3. Add Tseitin **extension**:  $p \leftrightarrow A$  (where  $p$  is a fresh atom).
4. Use the same sub-proof many times, via the **cut rule**.
5. Use the same sub-proof many times, in dag-ness, or **cocontraction**.

Only 5 is allowed in analytic proof systems.

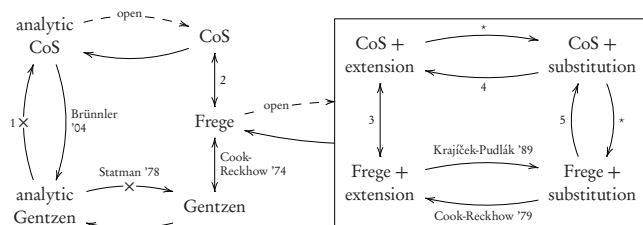
4 is the most studied form of compression.

## Compressing Proofs 2

Some facts:

- ▶ Substitution and extension are equivalent when added to Frege and to deep inference (not a trivial result).
- ▶ Any of these systems is usually called EF (for Extended Frege) and is considered the most interesting candidate as optimal proof system.
- ▶ The substitution/extension compression in deep inference leads to a bureaucracy-free formalism (but this is a topic for another talk).

# Proof Complexity and Deep Inference



Deep inference has as small proofs as the best systems (2,3,4,5,\*)  
and  
it has a normalisation theory  
and  
its analytic proof systems are more powerful than Gentzen ones (1)  
and  
cut elimination is  $n^{O(\log n)}$ , i.e., **quasipolynomial** (instead of exponential).

(See [Jeřábek(2009), Bruscoli & Guglielmi(2009), Bruscoli et al.(2009)Bruscoli, Guglielmi, Gundersen, & Parigot]).

# (Proof) System SKS

[Brünnler & Tiu(2001)]

- ▶ **Atomic** rules:

$\text{ai}\downarrow \frac{t}{a \vee \bar{a}}$	$\text{aw}\downarrow \frac{f}{a}$	$\text{ac}\downarrow \frac{a \vee a}{a}$
<i>identity</i>	<i>weakening</i>	<i>contraction</i>
$\text{ai}\uparrow \frac{a \wedge \bar{a}}{f}$	$\text{aw}\uparrow \frac{a}{t}$	$\text{ac}\uparrow \frac{a}{a \wedge a}$
<i>cut</i>	<i>coweakening</i>	<i>cocontraction</i>

- ▶ **Linear** rules:

$\text{s} \frac{A \wedge [B \vee C]}{(A \wedge B) \vee C}$	$\text{m} \frac{(A \wedge B) \vee (C \wedge D)}{[A \vee C] \wedge [B \vee D]}$
<i>switch</i>	<i>medial</i>

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

# Example 1

- ▶ In the calculus of structures (CoS):

$$\begin{array}{c}
 \text{ac}\uparrow \frac{[a \vee b] \wedge a}{[(a \wedge a) \vee b] \wedge a} \\
 \text{ac}\uparrow \frac{[(a \wedge a) \vee (b \wedge b)] \wedge a}{[(a \wedge a) \vee (b \wedge b)] \wedge (a \wedge a)} \\
 \text{ac}\uparrow \frac{[(a \wedge a) \vee (b \wedge b)] \wedge (a \wedge a)}{([a \vee b] \wedge [a \vee b]) \wedge (a \wedge a)} \\
 \text{m} \frac{([a \vee b] \wedge [a \vee b]) \wedge (a \wedge a)}{([a \vee b] \wedge a) \wedge ([a \vee b] \wedge a)}
 \end{array}$$

- ▶ In 'Formalism A':

$$\text{m} \frac{\frac{a}{a \wedge a} \vee \frac{b}{b \wedge b}}{[a \vee b] \wedge [a \vee b]} \wedge \frac{a}{a \wedge a}$$

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

## Example 2

► In CoS:

$$\begin{aligned}
 & \text{ai} \downarrow \frac{t}{a \vee \bar{a}} \\
 &= \frac{(a \wedge t) \vee (t \wedge \bar{a})}{\text{m} \frac{[a \vee t] \wedge [t \vee \bar{a}]}{[a \vee t] \wedge [\bar{a} \vee t]}} \\
 &= \frac{([a \vee t] \wedge \bar{a}) \vee t}{\text{s} \frac{(\bar{a} \wedge [a \vee t]) \vee t}{[(\bar{a} \wedge a) \vee t] \vee t}} \\
 &= \frac{(a \wedge \bar{a}) \vee t}{\text{ai} \uparrow \frac{f \vee t}{t}}
 \end{aligned}$$

► In 'Formalism A':

$$\begin{aligned}
 & \frac{t}{a \vee \bar{a}} \\
 & \text{m} \frac{[a \vee t] \wedge [t \vee \bar{a}]}{\text{s} \left[ \begin{array}{c} [a \vee t] \wedge \bar{a} \\ \frac{a \wedge \bar{a}}{f} \vee t \vee t \end{array} \right]}
 \end{aligned}$$

# Locality

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

Example, atomic cocontraction:

$$\frac{\frac{a}{a \wedge a} \vee \frac{b}{b \wedge b}}{[a \vee b] \wedge [a \vee b]} \wedge \frac{a}{a \wedge a}$$

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:

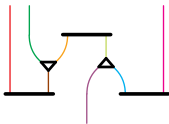
- ▶ **Deep inference** = locality (+ symmetry).
- ▶ **Locality** = atomicity + linearity.
- ▶ **Geometry** = syntax independence (elimination of bureaucracy) via **atomic flows**.
- ▶ We can also normalise in a **geometric** way.
- ▶ Locality (atomicity)  $\rightarrow$  geometry  $\rightarrow$  **semantics 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').



# (Atomic) Flows

$$\begin{array}{c}
 \frac{t}{a \vee \bar{a}} \\
 \frac{m}{[a \vee t] \wedge [t \vee \bar{a}]} \\
 \frac{s}{\left[ \frac{[a \vee t] \wedge \bar{a}}{\frac{a \wedge \bar{a}}{f} \vee t} \right]}
 \end{array}
 =
 \left(
 \begin{array}{c}
 a \wedge \left[ \frac{\bar{a} \vee \frac{t}{\bar{a} \vee a}}{\bar{a} \vee \bar{a}} \right] \\
 \frac{s}{\frac{a \wedge \bar{a}}{\bar{a}} \vee \frac{a}{a \wedge a}} \wedge \bar{a} \\
 \frac{f}{a \wedge \frac{a \wedge \bar{a}}{f}}
 \end{array}
 \right)
 \frac{m}{\frac{a}{a \wedge a} \vee \frac{b}{b \wedge b}} \wedge \frac{a}{a \wedge a}$$



- ▶ 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:

$$\text{aw}\downarrow\text{-ac}\downarrow: \begin{array}{c} \nabla \\ \swarrow \quad \searrow \\ \quad \nabla \\ \quad | \\ \quad 1,2 \end{array} \rightarrow \begin{array}{c} | \\ 1,2 \end{array}$$

$$\text{ac}\uparrow\text{-aw}\uparrow: \begin{array}{c} | \\ \nabla \\ \swarrow \quad \searrow \\ \quad \nabla \\ \quad | \\ \quad 1,2 \end{array} \rightarrow \begin{array}{c} | \\ 1,2 \end{array}$$

$$\text{aw}\downarrow\text{-ai}\uparrow: \begin{array}{c} \nabla \\ | \\ \hline \quad | \\ \quad 1 \\ \quad \nabla \end{array} \rightarrow \begin{array}{c} \nabla \\ | \\ \quad 1 \end{array}$$

$$\text{ai}\downarrow\text{-aw}\uparrow: \begin{array}{c} \hline \quad | \\ \quad 1 \\ \quad \nabla \end{array} \rightarrow \begin{array}{c} \nabla \\ | \\ \quad 1 \end{array}$$

$$\text{aw}\downarrow\text{-aw}\uparrow: \begin{array}{c} \nabla \\ | \\ \nabla \end{array} \rightarrow$$

$$\text{aw}\downarrow\text{-ac}\uparrow: \begin{array}{c} \nabla \\ \swarrow \quad \searrow \\ \quad \nabla \\ \quad | \\ \quad 1 \quad 2 \end{array} \rightarrow \begin{array}{c} \nabla \\ | \\ \quad 1 \end{array} \quad \begin{array}{c} \nabla \\ | \\ \quad 2 \end{array}$$

$$\text{ac}\downarrow\text{-aw}\uparrow: \begin{array}{c} \swarrow \quad \searrow \\ \quad \nabla \\ \quad | \\ \quad 1 \quad 2 \end{array} \rightarrow \begin{array}{c} | \\ \nabla \\ \quad 1 \end{array} \quad \begin{array}{c} | \\ \nabla \\ \quad 2 \end{array}$$

Each of them corresponds to a correct derivation reduction.

# Flow Reductions: (Co)Weakening (2)

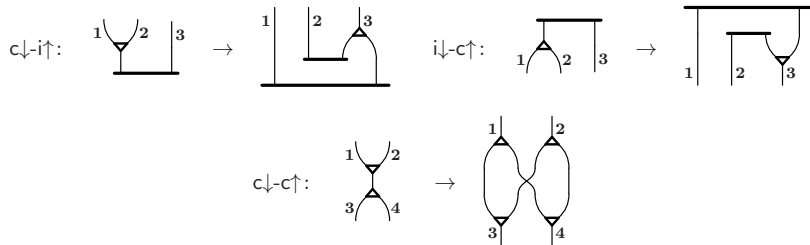
For example,  $\text{ai}\downarrow\text{-aw}\uparrow$ :   $\rightarrow$   specifies that

$$\begin{array}{ccc}
 \begin{array}{c}
 \Pi'' \parallel \\
 \xi \left\{ \frac{t}{a^\epsilon \vee \bar{a}} \right\} \\
 \Phi \parallel \\
 \zeta \left\{ \frac{a^\epsilon}{t} \right\} \\
 \Psi \parallel \\
 \alpha
 \end{array}
 & \text{becomes} &
 \begin{array}{c}
 \Pi'' \parallel \\
 \xi \left[ t \vee \frac{f}{\bar{a}} \right] \\
 \Phi_{\{a^\epsilon/t\}} \parallel \\
 \zeta \{t\} \\
 \Psi \parallel \\
 \alpha
 \end{array}
 \end{array}$$

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

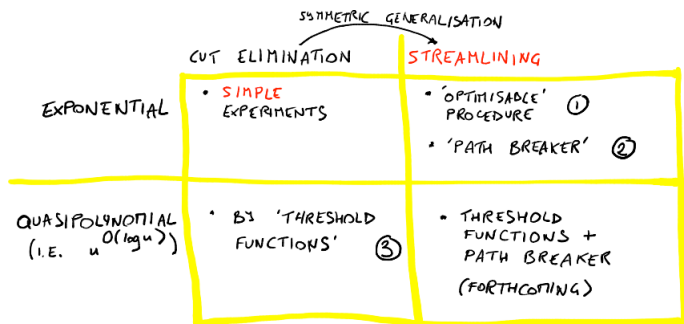
# Flow Reductions: (Co)Contraction

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

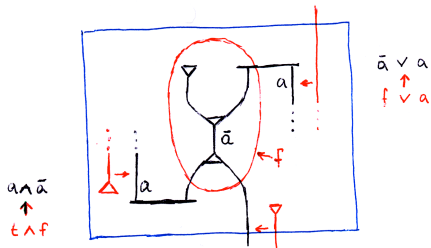


- ▶ None of these methods existed before atomic flows, none of them requires permutations or other syntactic devices.
- ▶ **Quasipolynomial** procedures are **surprising**.

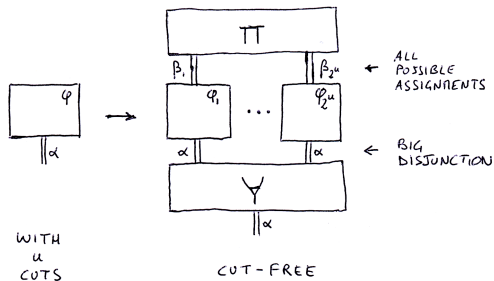
(1) [Guglielmi & Gundersen(2008)]; (2) LICS 2010 submission; (3) [Bruscoli et al.(2009)Bruscoli, Guglielmi, Gundersen, & Parigot].

# Cut Elimination (on Proofs) by 'Experiments'

Experiment:



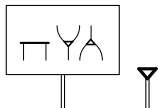
We do:



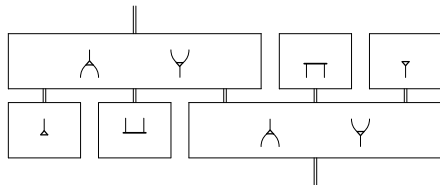
Simple, exponential cut elimination; proof generates  $2^n$  experiments.

# Generalising the Cut-Free Form

- ▶ Normalised proof:



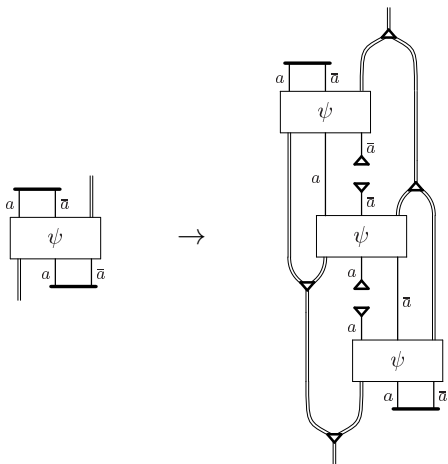
- ▶ Normalised derivation:



- ▶ The symmetric form is called **streamlined**.
- ▶ Cut elimination is a corollary of streamlining.
- ▶ We need to **break paths** between identity and cut nodes.

# 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

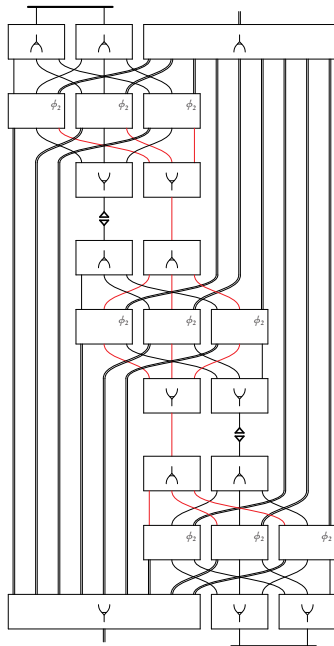
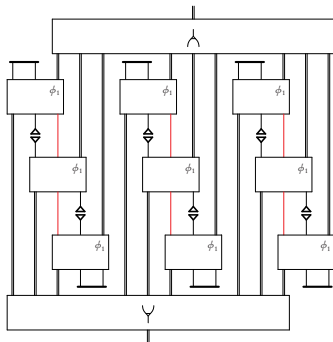
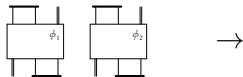
$$\frac{\frac{A}{[a \vee \bar{a}] \wedge A} \quad \Psi}{B \vee (a \wedge \bar{a})} \quad B$$

→

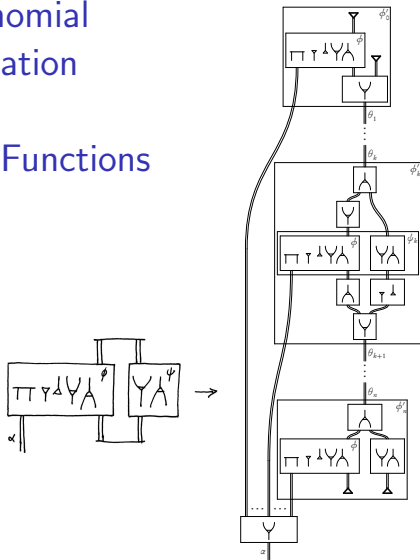
$$\frac{\frac{\frac{\frac{\frac{\frac{A}{([a \vee \bar{a}] \wedge A) \wedge A} \quad (\Psi \wedge A) \wedge A}{([B \vee (a \wedge \bar{a})] \wedge A) \wedge A} \quad \Phi_a \wedge A}{[B \vee ((a \vee \bar{a}) \wedge A)] \wedge A} \quad [B \vee \Psi] \wedge A}{B \vee ([B \vee (a \wedge \bar{a})] \wedge A)} \quad B \vee \Phi_a}{B \vee [B \vee ((a \vee \bar{a}) \wedge A)]} \quad B \vee [B \vee \Psi]}{B \vee [B \vee [B \vee (a \wedge \bar{a})]]} \quad \{c\downarrow, ai\uparrow, =\}$$

- ▶ 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

- ▶  $\theta_i =$  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  $\theta_i$  can be kind of projected into each atom to provide its **pseudocomplement**, for example the pseudocomplement of  $a$  in  $\theta_1$  is  $b$ .
- ▶ The atom and the pseudocomplement fit into the scheme of the previous slide, and you can get, for example,  $\theta_2$  from  $\theta_1$ .
- ▶ Stitch derivations together until you get  $\theta_{n+1} = f$ .
- ▶ The complexity is dominated by the complexity of the  $\theta$ 's, which is  $n^{O(\log n)}$ .

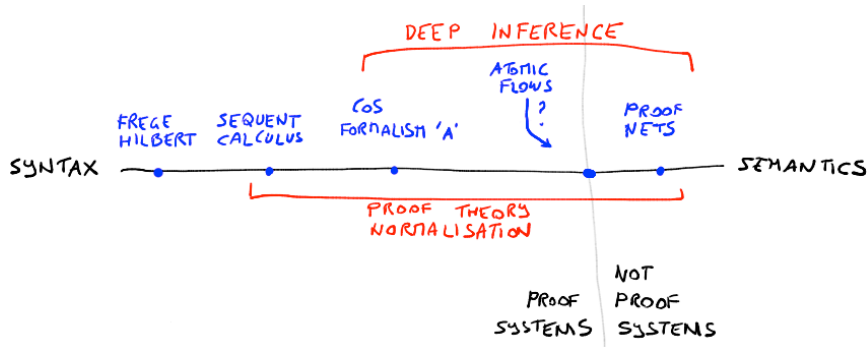
The difficulty is in defining the  $\theta$ 's and in finding proofs that stitch them together (this theory comes from circuit complexity and it had been applied to the monotone sequent calculus, which is weaker than propositional logic).

# Conjecture 1

We can normalise in polynomial time, because:

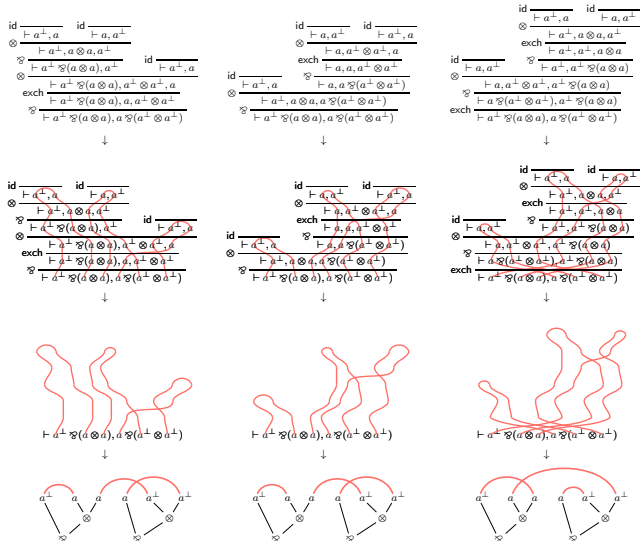
- ▶ polynomial threshold function representations exist;
- ▶ deep inference is flexible.

# Elimination of Bureaucracy



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

# State of the Art



From syntactically different proofs we obtain **proof nets**. They help, but they lose too much information (technically, they **do not form a proof system**).

# What Do We Need to Solve the Proof Identity Problem?

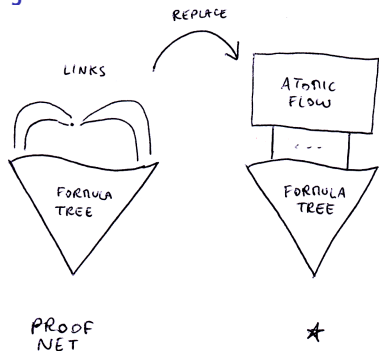
A finer representation of proofs, achieving **locality**.

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).



## 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  $\text{coNP} = \text{NP}$ .

# 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 analyticity 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/TYDI.pdf>

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/PrCompDI.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>.

Cook, S., & Reckhow, R. (1974).

**On the lengths of proofs in the propositional calculus (preliminary version).**  
In *Proceedings of the 6th annual ACM Symposium on Theory of Computing*, (pp. 135–148). ACM Press.

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>.