Removing Syntax From Proof Theory

Alessio Guglielmi

University of Bath

Joint work with
Paola Bruscoli, Tom Gundersen, Michel Parigot and Lutz Straßburger

9 December 2010

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

Outline

Proof Systems

- **Proof system** = algorithm checking proofs in polytime.

Proof Systems

- Proof system = algorithm checking proofs in polytime.
- Example, a Frege system:
  - Axioms:
    $$\begin{gathered} A \supset (B \supset A), \\ (A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C)), \\ (\neg B \supset \neg A) \supset ((\neg B \supset A) \supset B), \end{gathered}$$
    and rules (often just modus ponens, or cut): $\dfrac{A \quad A \supset B}{B}$.
  - $$\cfrac{\cfrac{}{a \supset (a \supset a)} \qquad \cfrac{\cfrac{}{a \supset ((a \supset a) \supset a)} \quad \cfrac{}{(a \supset ((a \supset a) \supset a)) \supset ((a \supset (a \supset a)) \supset (a \supset a))}}{(a \supset (a \supset a)) \supset (a \supset a)}}{a \supset a}.$$

Proof Systems

- Proof system = algorithm checking proofs in polytime.
- Example, a Frege system:
  - Axioms:
    $$A \supset (B \supset A),$$
    $$(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C)),$$
    $$(\neg B \supset \neg A) \supset ((\neg B \supset A) \supset B),$$

    and rules (often just modus ponens, or cut): $\dfrac{A \quad A \supset B}{B}$.
  - $$\dfrac{\overline{a \supset (a \supset a)} \qquad \dfrac{\overline{a \supset ((a \supset a) \supset a)} \qquad \overline{(a \supset ((a \supset a) \supset a)) \supset ((a \supset (a \supset a)) \supset (a \supset a))}}{(a \supset (a \supset a)) \supset (a \supset a)}}{a \supset a}.$$
- Robustness Theorem [Cook and Reckhow, 1974]:
  All Frege systems are polynomially equivalent.

Proof Systems

- Proof system = algorithm checking proofs in polytime.
- Example, a Frege system:
  - Axioms:
    $$A \supset (B \supset A),$$
    $$(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C)),$$
    $$(\neg B \supset \neg A) \supset ((\neg B \supset A) \supset B),$$

    and rules (often just modus ponens, or cut): $$\frac{A \quad A \supset B}{B}.$$
  - $$\dfrac{\overline{a \supset (a \supset a)} \qquad \dfrac{\overline{a \supset ((a \supset a) \supset a)} \quad \overline{(a \supset ((a \supset a) \supset a)) \supset ((a \supset (a \supset a)) \supset (a \supset a))}}{(a \supset (a \supset a)) \supset (a \supset a)}}{a \supset a}.$$
- Robustness Theorem [Cook and Reckhow, 1974]:
  All Frege systems are polynomially equivalent.
- Due to implicational completeness: if $A \supset B$ then A proves B.

Proof Complexity and the NP Vs. co-NP Problem

- Theorem [Cook and Reckhow, 1974]:

  $\exists$ *super proof system*
  iff
  NP = co-NP

  where

  super = with polysize proofs over the proved tautology

Proof Complexity and the NP Vs. co-NP Problem

- Theorem [Cook and Reckhow, 1974]:

  $\exists$ *super proof system*
  iff
  NP $=$ co-NP

  where

  super $=$ with polysize proofs over the proved tautology

- $\exists$ super proof system? Probably not; hard.

Proof Complexity and the NP Vs. co-NP Problem

- Theorem [Cook and Reckhow, 1974]:

  $\exists$ *super proof system*
  iff
  NP $=$ co-NP

  where

  super $=$ with polysize proofs over the proved tautology

- $\exists$ super proof system? Probably not; hard.
- $\exists$ optimal (polynomially simulating all others) proof system? 50/50; perhaps feasible.

Compressing Proofs

How can we make proofs smaller?

Compressing Proofs

How can we make proofs smaller?

Known mechanisms:

1. Higher orders (*e.g*, second order propositional for propositional formulae).
2. Tseitin extension: $p \leftrightarrow A$ (where p is a fresh atom).
3. Substitution: $\mathsf{sub}\,\dfrac{A}{A\sigma}$.
4. Use the same sub-proof many times: dag-ness, or cocontraction.
5. Use the same sub-proof many times: cut rule.

Compressing Proofs

How can we make proofs smaller?

Known mechanisms:

1. Higher orders (*e.g*, second order propositional for propositional formulae).
2. Tseitin extension: $p \leftrightarrow A$ (where p is a fresh atom).
3. Substitution: $\mathsf{sub}\,\dfrac{A}{A\sigma}$. Equivalent to (2).
4. Use the same sub-proof many times: dag-ness, or cocontraction.
5. Use the same sub-proof many times: cut rule.

Compressing Proofs

How can we make proofs smaller?

Known mechanisms:

1. Higher orders (*e.g*, second order propositional for propositional formulae).
2. Tseitin extension: $p \leftrightarrow A$ (where p is a fresh atom). Optimal?
3. Substitution: $\mathsf{sub}\,\frac{A}{A\sigma}$. Equivalent to (2).
4. Use the same sub-proof many times: dag-ness, or cocontraction.
5. Use the same sub-proof many times: cut rule.

Compressing Proofs

How can we make proofs smaller?

Known mechanisms:

1. Higher orders (*e.g*, second order propositional for propositional formulae).
2. Tseitin extension: $p \leftrightarrow A$ (where p is a fresh atom). Optimal?
3. Substitution: $\mathsf{sub}\frac{A}{A\sigma}$. Equivalent to (2).
4. Use the same sub-proof many times: dag-ness, or cocontraction.
5. Use the same sub-proof many times: cut rule.
   Most studied, proof theory.

Summary: Where Is Syntax?

Not in the notion of proof system:

- it's any algorithm with certain properties;
- Frege is robust.

Summary: Where Is Syntax?

Not in the notion of proof system:

- it's any algorithm with certain properties;
- Frege is robust.

Not in the compression mechanisms (higher orders, extension/substitution, cocontraction) . . .

Summary: Where Is Syntax?

Not in the notion of proof system:

- it's any algorithm with certain properties;
- Frege is robust.

Not in the compression mechanisms (higher orders, extension/substitution, cocontraction) . . .

. . . except for the cut and cut elimination (*i.e.*, Gentzen's proof theory).

Summary: Where Is Syntax?

Not in the notion of proof system:

- it's any algorithm with certain properties;
- Frege is robust.

Not in the compression mechanisms (higher orders, extension/substitution, cocontraction) . . .

. . . except for the cut and cut elimination (*i.e.*, Gentzen's proof theory).

So:

1. Can we capture cut and analyticity independently of syntax?

Summary: Where Is Syntax?

Not in the notion of proof system:

- it's any algorithm with certain properties;
- Frege is robust.

Not in the compression mechanisms (higher orders, extension/substitution, cocontraction) . . .

. . . except for the cut and cut elimination (*i.e.*, Gentzen's proof theory).

So:

1. Can we capture cut and analyticity independently of syntax?
2. Robustness?

Summary: Where Is Syntax?

Not in the notion of proof system:

- it's any algorithm with certain properties;
- Frege is robust.

Not in the compression mechanisms (higher orders, extension/substitution, cocontraction) . . .

. . . except for the cut and cut elimination (*i.e.*, Gentzen's proof theory).

So:

1. Can we capture cut and analyticity independently of syntax?
2. Robustness?

This talk answers YES to Question (1).

(Proof) System SKS [Brünnler and Tiu, 2001]

- Atomic rules:

$$\boxed{\begin{array}{ccc}
\mathsf{ai}{\downarrow}\dfrac{\mathsf{t}}{a \vee \bar{a}} & \mathsf{aw}{\downarrow}\dfrac{\mathsf{f}}{a} & \mathsf{ac}{\downarrow}\dfrac{a \vee a}{a} \\
\textit{identity} & \textit{weakening} & \textit{contraction} \\
\\
\mathsf{ai}{\uparrow}\dfrac{a \wedge \bar{a}}{\mathsf{f}} & \mathsf{aw}{\uparrow}\dfrac{a}{\mathsf{t}} & \mathsf{ac}{\uparrow}\dfrac{a}{a \wedge a} \\
\textit{cut} & \textit{coweakening} & \textit{cocontraction}
\end{array}}$$

(Proof) System SKS [Brünnler and Tiu, 2001]

- Atomic rules:

$$\begin{array}{ccc}
\mathsf{ai}{\downarrow}\,\dfrac{\mathsf{t}}{a \vee \bar{a}} & \mathsf{aw}{\downarrow}\,\dfrac{\mathsf{f}}{a} & \mathsf{ac}{\downarrow}\,\dfrac{a \vee a}{a} \\
\textit{identity} & \textit{weakening} & \textit{contraction} \\
\mathsf{ai}{\uparrow}\,\dfrac{a \wedge \bar{a}}{\mathsf{f}} & \mathsf{aw}{\uparrow}\,\dfrac{a}{\mathsf{t}} & \mathsf{ac}{\uparrow}\,\dfrac{a}{a \wedge a} \\
\textit{cut} & \textit{coweakening} & \textit{cocontraction}
\end{array}$$

- Linear rules:

$$\begin{array}{cc}
\mathsf{s}\,\dfrac{A \wedge [B \vee C]}{(A \wedge B) \vee C} & \mathsf{m}\,\dfrac{(A \wedge B) \vee (C \wedge D)}{[A \vee C] \wedge [B \vee D]} \\
\textit{switch} & \textit{medial}
\end{array}$$

(Proof) System SKS [Brünnler and Tiu, 2001]

- Atomic rules:

$$\mathsf{ai}{\downarrow}\frac{\mathsf{t}}{a \vee \bar{a}} \qquad \mathsf{aw}{\downarrow}\frac{\mathsf{f}}{a} \qquad \mathsf{ac}{\downarrow}\frac{a \vee a}{a}$$

identity *weakening* *contraction*

$$\mathsf{ai}{\uparrow}\frac{a \wedge \bar{a}}{\mathsf{f}} \qquad \mathsf{aw}{\uparrow}\frac{a}{\mathsf{t}} \qquad \mathsf{ac}{\uparrow}\frac{a}{a \wedge a}$$

cut *coweakening* *cocontraction*

- Linear 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).

(Proof) System SKS [Brünnler and Tiu, 2001]

- Atomic rules:

$$\begin{array}{ccc} \mathsf{ai}{\downarrow}\,\dfrac{\mathsf{t}}{a \vee \bar{a}} & \mathsf{aw}{\downarrow}\,\dfrac{\mathsf{f}}{a} & \mathsf{ac}{\downarrow}\,\dfrac{a \vee a}{a} \\ \textit{identity} & \textit{weakening} & \textit{contraction} \\ \mathsf{ai}{\uparrow}\,\dfrac{a \wedge \bar{a}}{\mathsf{f}} & \mathsf{aw}{\uparrow}\,\dfrac{a}{\mathsf{t}} & \mathsf{ac}{\uparrow}\,\dfrac{a}{a \wedge a} \\ \textit{cut} & \textit{coweakening} & \textit{cocontraction} \end{array}$$

- Linear rules:

$$\begin{array}{cc} \mathsf{s}\,\dfrac{A \wedge [B \vee C]}{(A \wedge B) \vee C} & \mathsf{m}\,\dfrac{(A \wedge B) \vee (C \wedge D)}{[A \vee C] \wedge [B \vee D]} \\ \textit{switch} & \textit{medial} \end{array}$$

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

(Proof) System SKS [Brünnler and Tiu, 2001]

- Atomic rules:

$$\boxed{\begin{array}{ccc}
\mathsf{ai}\!\downarrow \dfrac{\mathsf{t}}{a \vee \bar{a}} & \mathsf{aw}\!\downarrow \dfrac{\mathsf{f}}{a} & \mathsf{ac}\!\downarrow \dfrac{a \vee a}{a} \\
\textit{identity} & \textit{weakening} & \textit{contraction} \\[2ex]
\mathsf{ai}\!\uparrow \dfrac{a \wedge \bar{a}}{\mathsf{f}} & \mathsf{aw}\!\uparrow \dfrac{a}{\mathsf{t}} & \mathsf{ac}\!\uparrow \dfrac{a}{a \wedge a} \\
\textit{cut} & \textit{coweakening} & \textit{cocontraction}
\end{array}}$$

- Linear rules:

$$\boxed{\begin{array}{cc}
\mathsf{s}\,\dfrac{A \wedge [B \vee C]}{(A \wedge B) \vee C} & \mathsf{m}\,\dfrac{(A \wedge B) \vee (C \wedge D)}{[A \vee C] \wedge [B \vee D]} \\
\textit{switch} & \textit{medial}
\end{array}}$$

- Plus an ‘$=$’ linear rule (associativity, commutativity, units).
- Negation on atoms only.
- Cut is atomic.

(Proof) System SKS [Brünnler and Tiu, 2001]

- Atomic rules:

$$\begin{array}{ccc}
\text{ai}{\downarrow}\,\dfrac{\mathsf{t}}{a \vee \bar{a}} & \text{aw}{\downarrow}\,\dfrac{\mathsf{f}}{a} & \text{ac}{\downarrow}\,\dfrac{a \vee a}{a} \\
\textit{identity} & \textit{weakening} & \textit{contraction} \\[2ex]
\text{ai}{\uparrow}\,\dfrac{a \wedge \bar{a}}{\mathsf{f}} & \text{aw}{\uparrow}\,\dfrac{a}{\mathsf{t}} & \text{ac}{\uparrow}\,\dfrac{a}{a \wedge a} \\
\textit{cut} & \textit{coweakening} & \textit{cocontraction}
\end{array}$$

- Linear rules:

$$\begin{array}{cc}
\text{s}\,\dfrac{A \wedge [B \vee C]}{(A \wedge B) \vee C} & \text{m}\,\dfrac{(A \wedge B) \vee (C \wedge D)}{[A \vee C] \wedge [B \vee D]} \\
\textit{switch} & \textit{medial}
\end{array}$$

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

Examples in Open Deduction

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

Examples in Open Deduction

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

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

Examples in Open Deduction

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

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

Proofs are composed by the same operators as formulae.

Examples in Open Deduction

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

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

Proofs are composed by the same operators as formulae.

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

Examples in Open Deduction

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

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

Proofs are composed by the same operators as formulae.

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

(In [Guglielmi et al., 2010a].)

Locality

Deep inference allows locality,

i.e.,

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

Locality

Deep inference allows locality,

i.e.,

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

E.g., atomic cocontraction:

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

Locality

Deep inference allows locality,

i.e.,

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

E.g., atomic cocontraction:

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

In Gentzen:

- no locality for (co)contraction (counterexample in [Brünnler, 2004]),
- no local reduction of cut into atomic form.

Slogans

Deep inference $=$ locality $(+$ symmetry$)$.

Slogans

Deep inference $=$ locality $(+$ symmetry$)$.

Locality $=$ linearity $+$ atomicity.

Slogans

Deep inference $=$ locality ($+$ symmetry).

Locality $=$ linearity $+$ atomicity.

Geometry $=$ syntax independence (elimination of bureaucracy).

Slogans

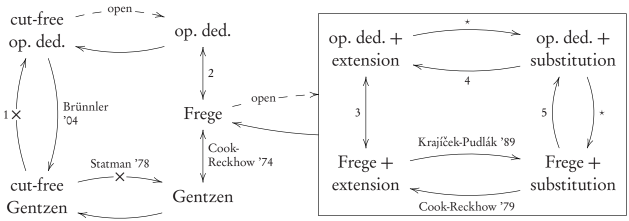
Deep inference $=$ locality $(+$ symmetry$)$.

Locality $=$ linearity $+$ atomicity.

Geometry $=$ syntax independence (elimination of bureaucracy).

Locality $\rightarrow$ geometry $\rightarrow$ semantics of proofs.

Deep Inference and Proof Complexity



cut-free op. ded.

open

op. ded.

2

1 ×

Brünnler '04

Frege

open

Cook-Reckhow '74

cut-free Gentzen

Statman '78

×

Gentzen

op. ded. + extension

⋆

op. ded. + substitution

4

3

5

⋆

Frege + extension

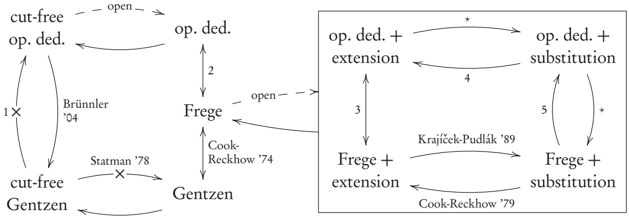
Krajíček-Pudlák '89

Frege + substitution

Cook-Reckhow '79

$\longrightarrow =$ 'polynomially simulates'.

Deep Inference and Proof Complexity



cut-free op. ded.

open

op. ded.

2

1 ×

Brünnler '04

Frege

open

Cook-Reckhow '74

Statman '78

cut-free Gentzen

Gentzen

op. ded. + extension

⋆

op. ded. + substitution

4

3

5

⋆

Krajíček-Pudlák '89

Frege + extension

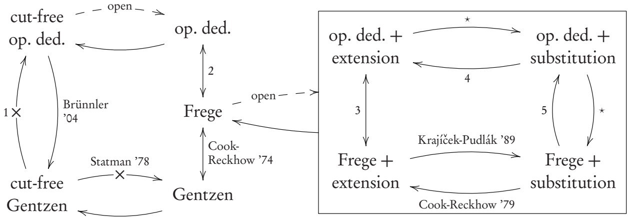
Frege + substitution

Cook-Reckhow '79

$\longrightarrow =$ 'polynomially simulates'.

Open deduction has as small proofs as the best formalisms

Deep Inference and Proof Complexity



cut-free op. ded.

open

op. ded.

1 ×

Brünner '04

2

Frege

open

Statman '78

Cook-Reckhow '74

cut-free Gentzen

Gentzen

op. ded. + extension

⋆

op. ded. + substitution

4

3

5

⋆

Krajíček-Pudlák '89

Frege + extension

Frege + substitution

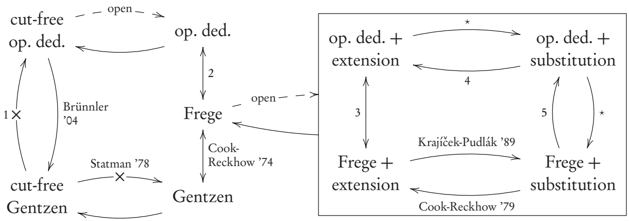
Cook-Reckhow '79

$\longrightarrow =$ 'polynomially simulates'.

Open deduction has as small proofs as the best formalisms and

it has a normalisation theory

Deep Inference and Proof Complexity



$\longrightarrow =$ 'polynomially simulates'.

Open deduction has as small proofs as the best formalisms

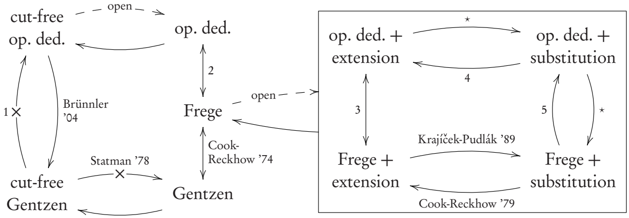
and

it has a normalisation theory

and

its cut-free proof systems are more powerful than Gentzen ones

Deep Inference and Proof Complexity



$\longrightarrow =$ 'polynomially simulates'.

Open deduction has as small proofs as the best formalisms

and

it has a normalisation theory

and

its cut-free proof systems are more powerful than Gentzen ones

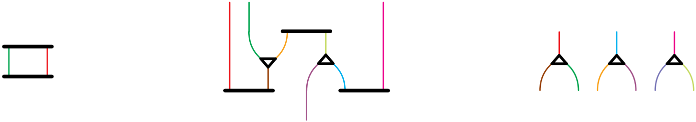
and

cut elimination is quasipolynomial (instead of exponential).

(See [Jeřábek, 2009, Bruscoli and Guglielmi, 2009, Bruscoli et al., 2010]).

(Atomic) Flows

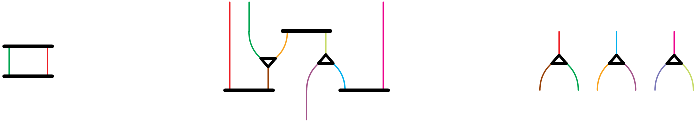
$$
{\scriptstyle\mathsf{s}}\,\frac{{\scriptstyle\mathsf{m}}\,\dfrac{\dfrac{\mathsf{t}}{a \vee \bar{a}}}{[a \vee \mathsf{t}] \wedge [\mathsf{t} \vee \bar{a}]}}{\left[{\scriptstyle\mathsf{s}}\,\dfrac{[a \vee \mathsf{t}] \wedge \bar{a}}{\dfrac{a \wedge \bar{a}}{\mathsf{f}} \vee \mathsf{t}} \vee \mathsf{t}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



Below proofs, their (atomic) flows are shown:

(Atomic) Flows

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

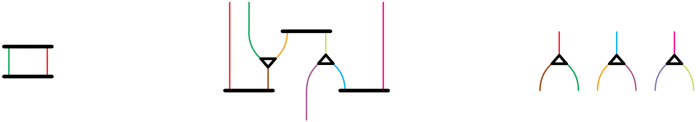


Below proofs, their (atomic) flows are shown:

- only structural information is retained in flows;

(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 proofs, their (atomic) flows are shown:

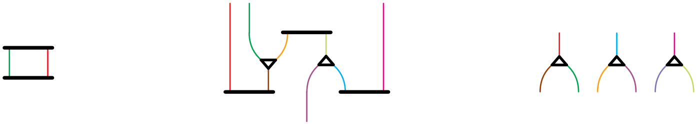
- only structural information is retained in flows;
- logical information is lost;

(Atomic) Flows

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

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

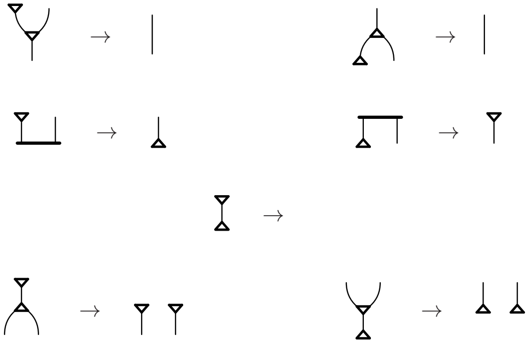
$$\mathsf{m}\,\dfrac{\dfrac{a}{a\wedge a}\vee\dfrac{b}{b\wedge b}}{[a\vee b]\wedge[a\vee b]}\quad\wedge\quad\dfrac{a}{a\wedge a}$$



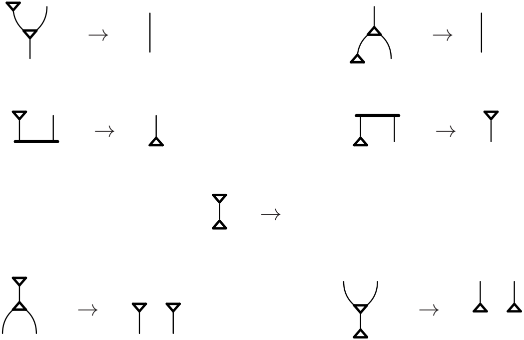
Below proofs, their (atomic) flows are shown:

- only structural information is retained in flows;
- logical information is lost;
- flow size is polynomially related to derivation size.

Flow Reductions: (Co)Weakening (1)



Flow Reductions: (Co)Weakening (1)



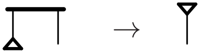
Each flow reduction corresponds to a correct proof reduction.

Flow Reductions: (Co)Weakening (2)

E.g.,  →  specifies that

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

Flow Reductions: (Co)Weakening (2)

E.g.,  specifies that

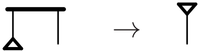
$$\begin{array}{c} \Pi''\Big\Vert \\ \xi\left\{\dfrac{\mathsf{t}}{a^{\epsilon}\vee\bar{a}}\right\} \\ \Phi\Big\Vert \\ \zeta\left\{\dfrac{a^{\epsilon}}{\mathsf{t}}\right\} \\ \Psi\Big\Vert \\ \alpha \end{array} \qquad \text{becomes} \qquad \begin{array}{c} \Pi''\Big\Vert \\ \xi\left[\mathsf{t}\vee\dfrac{\mathsf{f}}{\bar{a}}\right]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

We can operate on flow reductions instead than on derivations:

- much easier,

Flow Reductions: (Co)Weakening (2)

E.g.,



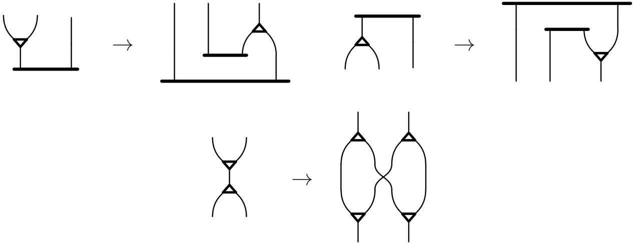
specifies that

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

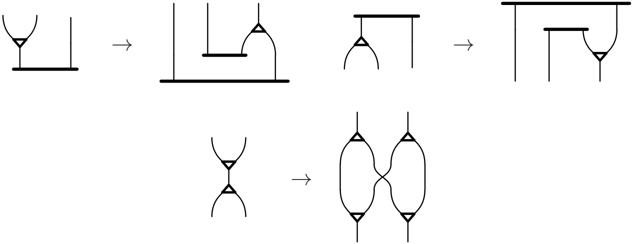
We can operate on flow reductions instead than on derivations:

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

Flow Reductions: (Co)Contraction

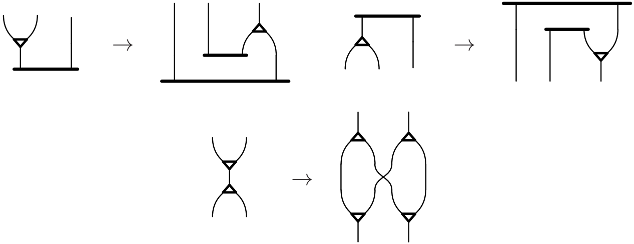


Flow Reductions: (Co)Contraction



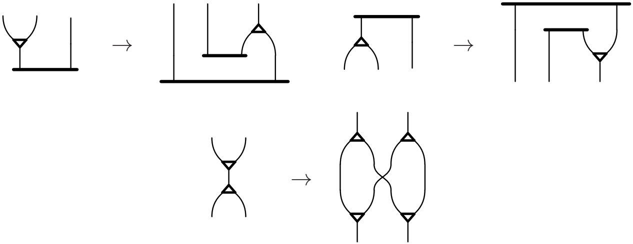
- These reductions conserve the number and length of paths.

Flow Reductions: (Co)Contraction



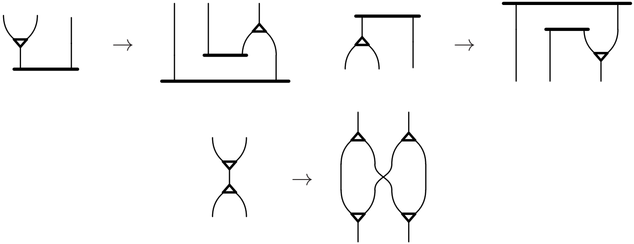
- These reductions conserve the number and length of paths.
- They can blow up a derivation exponentially.

Flow Reductions: (Co)Contraction



- These reductions conserve the number and length of paths.
- They can blow up a derivation exponentially.
- It's a good thing: cocontraction is a new compression mechanism (dag-ness?).

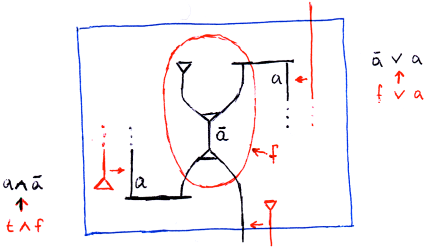
Flow Reductions: (Co)Contraction



- These reductions conserve the number and length of paths.
- They can blow up a derivation exponentially.
- It's a good thing: cocontraction is a new compression mechanism (dag-ness?).
- Open problem: does cocontraction yield exponential compression? Conjecture: yes.

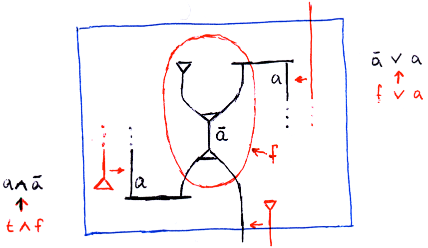
Cut Elimination by ‘Experiments’

Experiment:

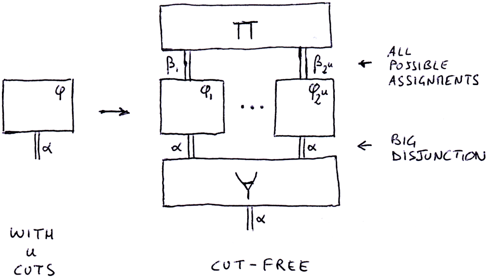


Cut Elimination by ‘Experiments’

Experiment:

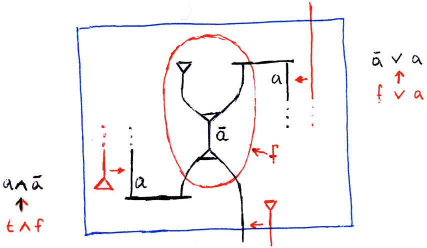


We do:

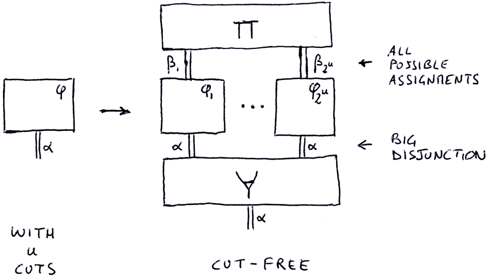


Cut Elimination by 'Experiments'

Experiment:



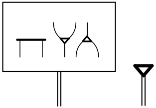
We do:



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

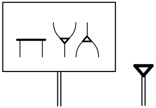
Generalising the Cut-Free Form

- Normalised proof:

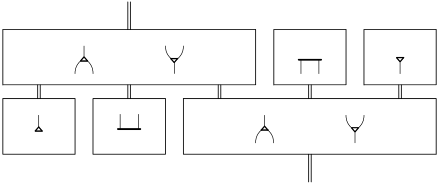


Generalising the Cut-Free Form

- Normalised proof:

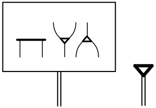


- Normalised derivation:

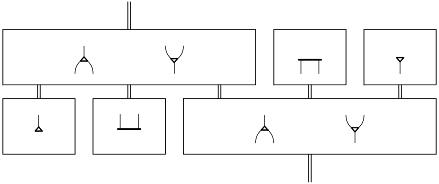


Generalising the Cut-Free Form

- Normalised proof:



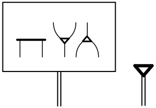
- Normalised derivation:



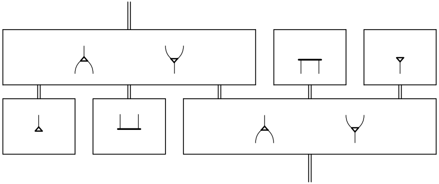
- The symmetric form is called streamlined.
- Cut elimination is a corollary of streamlining.

Generalising the Cut-Free Form

- Normalised proof:



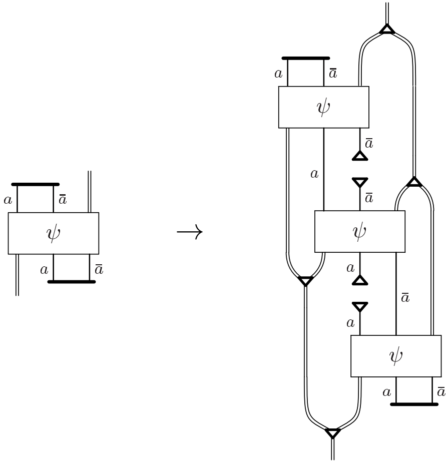
- Normalised derivation:



- 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 [Guglielmi et al., 2010b]:



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

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

We Can Do This on Derivations, of Course

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

- We can compose this as many times as there are paths between identities and cut.

We Can Do This on Derivations, of Course

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

- 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.

We Can Do This on Derivations, of Course

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

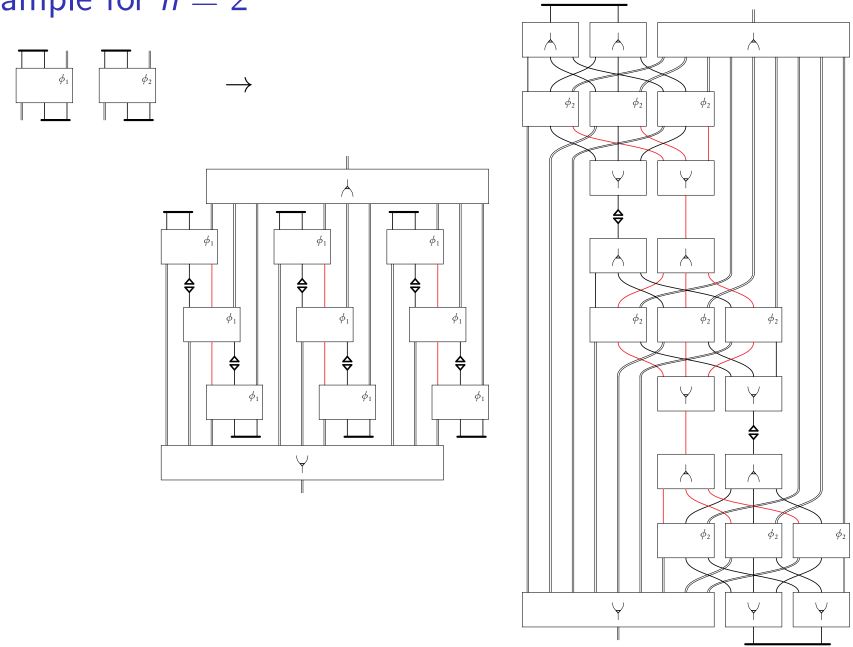
- We can compose this as many times as there are paths between identities and cut.
- We obtain a family of normalisers that only depends on n.
- The construction is exponential.

We Can Do This on Derivations, of Course

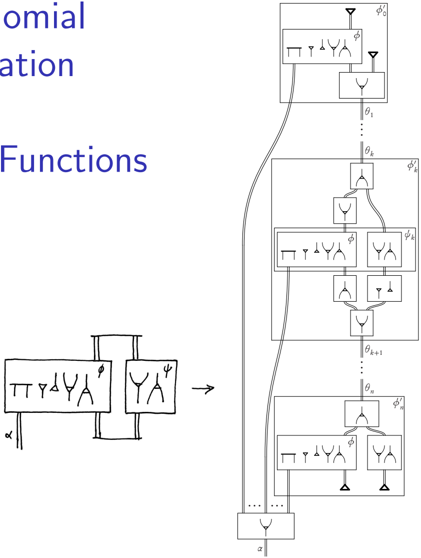
$$\begin{array}{c} A \\ \hline [a \vee \bar{a}] \wedge A \\ {\scriptstyle \Psi}\Big\| \\ B \vee (a \wedge \bar{a}) \\ \hline B \end{array} \qquad \rightarrow \qquad \begin{array}{c} A \\ \Big\|{\scriptstyle \{\mathsf{c}\uparrow, \mathsf{ai}\downarrow, =\}} \\ (([a \vee \bar{a}] \wedge A) \wedge A) \wedge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

- 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 $n = 2$

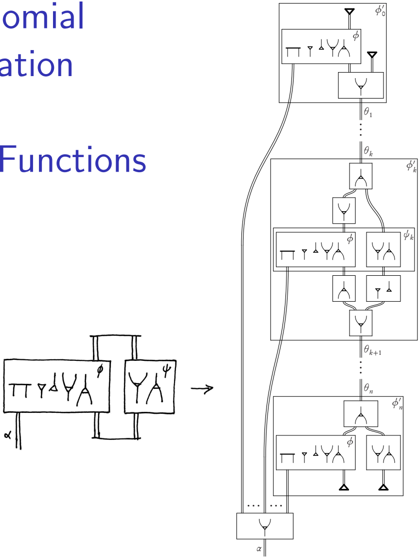


Quasipolynomial Cut Elimination by Threshold Functions



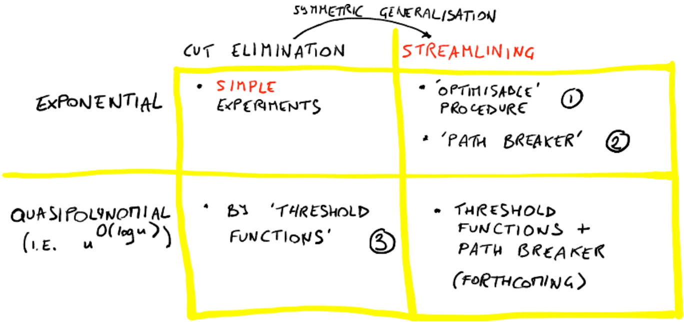
- Only $n+1$ copies of the proof are stitched together.

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).

Normalisation
Overview



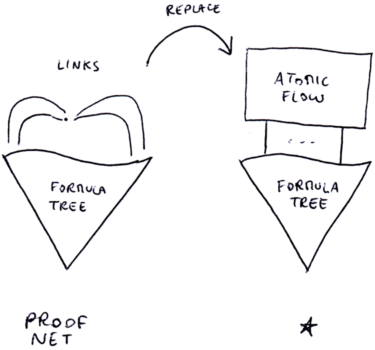
SYMMETRIC GENERALISATION

	CUT ELIMINATION	STREAMLINING
EXPONENTIAL	• SIMPLE EXPERIMENTS	• 'OPTIMISABLE' PROCEDURE ① • 'PATH BREAKER' ②
QUASIPOLYNOMIAL (I.E. $n^{O(\log n)}$)	• BY 'THRESHOLD FUNCTIONS' ③	• THRESHOLD FUNCTIONS + PATH BREAKER (FORTHCOMING)

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

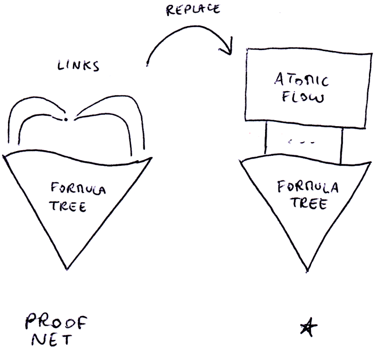
(1, 2) [Guglielmi et al., 2010b]; (3) [Bruscoli et al., 2010].

Conjecture



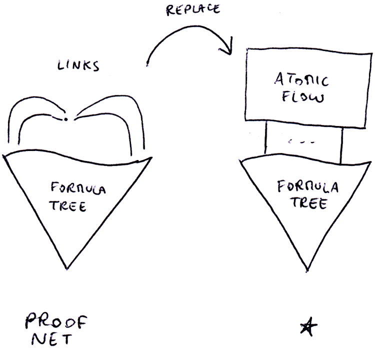
- We think that $*$ might make for a proof system.

Conjecture



- We think that $*$ might make for a proof system.
- If true, excellent bureaucracy-free formalism.

Conjecture



- We think that $*$ might make for a proof system.
- If true, excellent bureaucracy-free formalism.
- Note: if such a thing existed for proof nets, then coNP $=$ NP (because proof nets are [too?] small).

Conclusion

- Cut elimination does not depend on logical rules.
- It only depends on structural information, *i.e.*, geometry.
- Normalisation can be made robust.

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

References

Brünnler, K. (2004).

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

Brünnler, K. and Tiu, A. F. (2001).

A local system for classical logic.
In Nieuwenhuis, R. and Voronkov, A., editors, *LPAR 2001*, volume 2250 of *Lecture Notes in Computer Science*, pages 347–361. Springer-Verlag.
http://www.iam.unibe.ch/~kai/Papers/lcl-lpar.pdf.

Bruscoli, P. and 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., and Parigot, M. (2010).

A quasipolynomial cut-elimination procedure in deep inference via atomic flows and threshold formulae.
Accepted by LPAR-16. http://cs.bath.ac.uk/ag/p/QPNDI.pdf.

Cook, S. and 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*, pages 135–148. ACM Press.

Guglielmi, A., Gundersen, T., and Parigot, M. (2010a).

A proof calculus which reduces syntactic bureaucracy.
In Lynch, C., editor, *21st International Conference on Rewriting Techniques and Applications*, volume 6 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 135–150. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
http://drops.dagstuhl.de/opus/volltexte/2010/2649.

Guglielmi, A., Gundersen, T., and Straßburger, L. (2010b).

Breaking paths in atomic flows for classical logic.
In Jouannaud, J.-P., editor, *25th Annual IEEE Symposium on Logic in Computer Science*, pages 284–293. IEEE.
http://www.lix.polytechnique.fr/~lutz/papers/AFII.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.