Making Several Computation/Normalisation Mechanisms Coexist

Alessio Guglielmi (Bath)

Joint work with Andrea Aler Tubella
and Benjamin Ralph (both Bath)

Invited talk at FISP Kick-off 2016 (Innsbruck)
16/11/16

Talk available at
http://cs.bath.ac.uk/ag/t/MSCNMC.pdf
And from my home page
GentzenProof Theory

Compression? Manipulation? Locality? Modularity?...
... Names! Scope! Quantifiers! Substitutions! Trees!

In Gentzen:

< Formula > →

< Tautology >

The formula tree gives shape to the proof tree.

Good

Not good enough
Deep inference: better trees

A better proof tree looks like this:

- Vertical composition: inference
- Horizontal composition: any (!) connective

It's more general than Gentzen trees and yields:
- Exponential speed-ups
- Nonelementary
- Analyticity for modal and exotic logics
- Canonical forms
- Computational interpretations for concurrency

... and much more
**Compression and Naturality**

**Identity**

\[ \vdash P(z) \quad P(z) \]

\[ \vdash P(z) \quad P(z) \lor P(y) \]

\[ \vdash \neg P(x) \lor P(z) \quad P(z) \lor P(y) \]

\[ \vdash \neg P(x) \lor P(z) \quad \forall y \ (\neg P(z) \lor P(y)) \]

\[ \vdash \forall y \ (\neg P(x) \lor P(y)) \quad \exists x \forall y \ (\neg P(x) \lor P(y)) \]

\[ \vdash \exists x \forall y \ (\neg P(x) \lor P(y)) \]

**Weakening**

\[ \vdash P(x) \lor P(y) \]

\[ \forall y \ (\neg P(z) \lor P(y)) \]

\[ \exists x \forall y \ (\neg P(x) \lor P(y)) \]

\[ \vdash \exists x \forall y \ (\neg P(x) \lor P(y)) \]

**Contraction**

\[ \vdash \exists x \forall y \ (\neg P(x) \lor P(y)) \]

\[ \vdash \exists x \forall y \ (\neg P(x) \lor P(y)) \]

\[ \vdash \exists x \forall y \ (\neg P(x) \lor P(y)) \]

**Sequent Calculus**

**Deep Inference**

**Sound!**

---

**Apparently the speed-up is nonelementary**

*(See Aguilera-Braaž, we are checking the details)*
**Trivial Cut Elimination**

Assignment $A_\sigma = \alpha \land \beta \land \cdots \land \varphi \land \cdots$

\[
\begin{align*}
\text{Cut} & \quad \frac{B \land \bar{B}}{f} \\
\text{Apply } \sigma & \quad \frac{B_\sigma \land \bar{B}_\sigma}{f} = \frac{f \land \top}{f}
\end{align*}
\]

The cut disappears.

**Proof $\varphi$ with cuts**

Conclusion of $\varphi$ $C$

Apply $\sigma_1, \ldots, \sigma_n$

**Canonical proof with no cuts**

$A_{\sigma_1}$ $\varphi \sigma_1$

No cuts

$\cdots$

$\cdots$

$A_{\sigma_n}$ $\varphi \sigma_n$

No cuts

Several contractions

$C$

**Canonical Form!** Non-computational normalisation
Atomic Flows

\[
\begin{align*}
\alpha \land \frac{t}{\alpha \lor \bar{\beta}} \\
\alpha \lor (\alpha \land \bar{\beta}) \\
\alpha \land \alpha \\
\alpha \lor \alpha \\
\alpha \\
\alpha \land \bar{\beta} \\
\alpha \lor \bar{\beta} \\
\end{align*}
\]

Proof

Atomic Flow

There's enough information in an atomic flow to eliminate cuts in propositional classical logic.
Computation vs Normalisation in Deep Inference

Normalisation not necessarily computation

However

Several independent normalisation/computation mechanisms:

- **Cut Elimination** (Splitting): (Yanking) No complexity (Linear Logic)

- **Contraction** (Decomposition): (Unfolding) Exponential

- **Substitution** Exponential

- **Herbrand Expansion** Non-elementary?

In Gentzen these mechanisms are conflated:

$$
\Gamma, w A \vdash u \bar{A}, \Delta
\Rightarrow
\Gamma, \Delta
$$
INDEPENDENCE OF CUT ELIMINATION AND CONTRACTION

**Idea:**

**Problem:**
Cycle elimination 1

We will remove the green cycle by breaking its cut.

We move the contraction past the cut.
Cycle Elimination 2

We get a derivation whose red step is invalid.

Idea:
Cycle Elimination 3

We let the (co)weakenings go. This fixes the invalid step and simplifies the derivation.

Ok: complexity? We don't know yet.
Substitution

\[
\frac{a \land \bar{a}}{f} \left/ \left\{ \substack{a \leftarrow \frac{b \lor b}{b} \end{array} \right. \right. = \left\{ \frac{b \lor b}{b} \land \bar{b}}{f} \right. \right. \approx \left. \frac{(b \lor b) \land \frac{\bar{b}}{\bar{b} \land \bar{b}}}{f} \right. \right. \]

Exponential compression mechanism

Needs to be independent of

- decomposition OK
- splitting (linear cut elimination) \rightarrow
INDEPENDENCE OF CUT ELIMINATION AND SUBSTITUTION

\[ \text{WITH} \quad a \in A \]

\[ \text{CUT} \quad a \in B \]

\[ \text{FREE} \quad a \in C \]

\[ \text{NOT} \quad \text{CUT FREE!} \]

**BECAUSE**

\[ \text{T} \leftarrow \text{T} \text{T} = \{ \text{T} \text{T}, \text{T} \text{T}, \text{T} \text{T}, \text{T} \text{T} \} \]

**THIS DOES NOT WORK.**
INDEPENDENCE OF CUT ELIMINATION AND SUBSTITUTION 2

THIS WORKS.
Summary

- Gentzen's proof theory suffers from a massive artefact: a weak tree structure.

- Cut elimination in Gentzen is the conflation of two mechanisms: splitting and decomposition.

- There are several, independent normalisation mechanisms.
Nonelementary speed-up [1]; atomic flows [4, 5]; deep inference [2]; splitting-decomposition [3, 6, 7]. The rest is on manuscripts, available on request.


