Symmetric Normalisation for Intuitionistic Logic

Nicolas Guenot* & Lutz Straßburger
IT University of Copenhagen INRIA / Ecole Polytechnique
ngue@itu.dk lutz@lix.polytechnique.fr

CSL / LICS 2014, Vienna
What this talk is about:

**Normalisation** as known in Gentzen-style systems can be generalised in the **calculus of structures** to a **fully symmetric** transformation, using standard tools.

and slightly more generally:

**Normalisation** is a form of **decomposition**.
The calculus of structures in a nutshell:

Deduction proceeds by sound rewriting of one formula into another formula.

Consequences of this idea:

- derivations are not trees but sequences (no branching)
- a proof of $A$ is a derivation from a “truth” unit to $A$
- derivations are possible interleavings of proof trees
- each rule admits a sound dual rule (think contraposition)
- the meta-level operators $\vdash$ and $,$ are superfluous
Calculus of Structures (JS) vs. Sequent Calculus (LJ)

\[
\begin{align*}
\frac{\Delta}{(\Delta \supset A) \supset A} & \quad i \downarrow \\
\frac{((A \supset B) \supset C) \supset D}{A \supset ((B \supset C) \supset D)} & \quad s \downarrow \\
\frac{B}{A \supset B} & \quad w \downarrow \\
\frac{A \supset (A \supset B)}{A \supset B} & \quad c \downarrow \\
\frac{(\Delta \supset A) \supset A}{\Delta \supset B} & \quad i' \uparrow \\
\end{align*}
\]

\[
\begin{align*}
\frac{ax}{A \vdash A} & \quad tr \quad \vdash \top \\
\frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \supset B \vdash C} & \quad il \\
\frac{\Gamma, A \vdash B}{\Gamma, \Delta \vdash B} & \quad ir \\
\frac{\Gamma \vdash B}{\Gamma, A \vdash B} & \quad wk \\
\frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B} & \quad ct \\
\frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B} & \quad cut
\end{align*}
\]

where $\Delta \supset B$ is a notation for $A_1 \supset \cdots \supset A_n \supset B$, if $\Delta = \{A_1, \cdots, A_n\}$.

---

**Intuitionistic Logic in the Calculus of Structures**
A number of logics have been represented in deep inference:

- classical logic and intuitionistic logic
- linear logic and non-commutative variants
- various modal and hybrid logics
- linear logic with equality and fixpoints
- classical logic with extension or substitution
- bi-intuitionistic and tense logics
- and others...

...but possible computational interpretations of normalisation remain largely unexplored!

*Normalisation in the Calculus of Structures*
Cut elimination in a generalised setting:

- there is a direct equivalent of the \textit{cut} rule
- there is more \textit{symmetry} in the calculus of structures
- we cannot rely on the \textit{tree} structure of proofs
- many more \textit{permutations} appear in deep inference
- there is a more general notion of \textit{normalisation} there

Conclusion: we need either \textit{new} proof techniques or \textit{generalisations} of the known techniques.
Normalisation Proofs
in the Calculus of Structures
What is normalisation in the calculus of structures?

The **up fragment** of a system SZS, formed of the duals to the rules of ZS, is **admissible** in ZS.

Usual properties of such a deep inference system:

- **cut** is the dual of **identity**
- both cut and identity can trivially be made **atomic**
- all up rules can be **encoded** with cut and ZS rules
- derivations can be **plugged** inside other derivations
How can this be established?

- **Proof by soundness and completeness:**
  you have, for example, a sequent calculus LZ for the same logic, enjoying cut elimination, so just translate to LZ with cut and then back to ZS.

- **Proof by splitting and context reduction:**
  show that proofs can be splitted along connectives, then use this to prove that dual rules can be considered out of context, and show admissibility.

- **Proof by separation and substitution:**
  in classical logic, make cuts atomic and out of context and split a proof at a cut, one is broken but can be fixed by plugging the other into it.

- **Proof by permutation:**
  permute up rules upwards until they disappear, through interaction with corresponding down rules.
Intuitionistic Logic
in the Calculus of Structures
The basic syntax comprises formulas:

\[ A, B ::= a \mid \top \mid A \sqcup B \]

considered under a congruence to form structures:

\[ \top \sqcup A \equiv A \quad A \sqcup (B \sqcup C) \equiv B \sqcup (A \sqcup C) \]

and extended to positive and negative contexts:

\[ \pi ::= \lnot \mid \eta \sqcup A \quad \eta ::= \pi \sqcup A \]

Intuitionistic inference rules are sensitive to the polarity of the context where they are applied.
Here is the JS system, sound and complete for LJ, with atomic identity:

\[
\begin{align*}
&\frac{\pi \{\Delta\}}{\pi \{(\Delta \supset a) \supset a\}} & \frac{\pi \{B\}}{\pi \{A \supset B\}} \\
&\frac{\pi \{((A \supset B) \supset C) \supset D\}}{\pi \{A \supset (B \supset C) \supset D\}} & \frac{\pi \{A \supset A \supset B\}}{\pi \{A \supset B\}} \\
&\frac{i \uparrow \eta \{(\Delta \supset A) \supset A\}}{\eta \{\Delta\}}
\end{align*}
\]

As an example, consider the following derivations:

\[
\begin{align*}
\equiv & \quad \top \supset a \\
&\frac{\top}{\pi \{\Delta\}} \\
&\frac{a}{\pi \{B \supset a\}} \\
&\frac{((B \supset a) \supset a)}{\pi \{((B \supset a) \supset a) \supset a\}} \\
&\frac{(a \supset B) \supset (C \supset D \supset A) \supset B}{\equiv}
\end{align*}
\]
Cut Elimination
for Intuitionistic Logic
We want to use **permutations** to perform **yanking** on cuts:

![Diagram showing how to perform yanking on cuts using permutations.](image-url)
The required transformation comes with problems:

- as illustrated, there can be **contractions**
- a part of the given proof must be **duplicated**
- yanking works only on **atomic** cuts
- we need to track the **flow** of atoms

---

We will prove normalisation as follows:

1. trace the **flow-graph** of formulas in the proof
2. count the **contractions** affecting each of them
3. make cuts atomic using a **merging** lemma
4. move cuts upwards together **with subderivations**
A **flow-graph** is obtained by tracing atoms:

\[
((((a \triangleright a \triangleright B) \triangleright c) \triangleright a \triangleright B) \triangleright c) \triangleright D
\]

We can now count how many times a formula is used:

The **multiplicity** of a formula is the number of sources in the flow-graph of this formula.

For example, in the derivation above, \(a\) has multiplicity 3, and \((a \triangleright B) \triangleright c\) has multiplicity 2.
Atomic cuts are obtained through the merging lemma, which says:

A positive substructure in any given proof can be moved deep inside a positive context.

Reductions for atomic cut elimination are trivial permutations or cases involving erasure/duplication, or main cases such as:

\[
\begin{align*}
\text{ai} \downarrow & \quad \frac{\eta \{a\}}{\eta \{(a \supset a) \supset a\}} \\
\text{ai} \uparrow & \quad \frac{\eta \{a\}}{\eta \{a\}} \\
\end{align*}
\]

All atomic cuts can be eliminated this way.
Is this interesting in terms of computation?

This cut elimination procedure changes the shape of proofs in the global rewriting step of merging.

The atomic cut elimination is computational, but the problem lies in the reduction of cuts on an implication:

\[
\begin{align*}
\eta \{ (\Delta \supset A \supset B) \supset A \supset B \} \\
\eta \{ \Delta \} \\
\end{align*}
\]  
\begin{align*}
\stackrel{i \uparrow}{=} \frac{\eta \{ (\Delta \supset A \supset B) \supset A \supset B \}}{\eta \{ \Delta \}} \\
\end{align*}

\[
\begin{align*}
\eta \{ A \supset (\Delta \supset A \supset B) \supset B \} \\
\eta \{ (\Delta \supset (A \supset A) \supset B) \supset B \} \\
\eta \{ (\Delta \supset B) \supset B \} \\
\eta \{ \Delta \} \\
\end{align*}
\]  
\begin{align*}
\stackrel{\text{merging} \ i \uparrow}{=} \frac{\eta \{ (\Delta \supset A \supset B) \supset A \supset B \}}{\eta \{ \Delta \}} \\
\end{align*}

Cut Elimination for Intuitionistic Logic
But it so happens that the rule implemented by merging:

\[
\frac{\eta\{A \supset (B \supset C) \supset D\}}{\eta\{((A \supset B) \supset C) \supset D\}}
\]

is exactly the dual of the switch rule from JS!

As usual, cut can be made atomic in a symmetric system, but then a proof must be normalised with respect to all rules of the up fragment, not just cut.
Symmetric Normalisation
for Intuitionistic Logic
JS can be completed into the **symmetric system SJS**:

\[
\begin{align*}
\text{ai} & \quad \frac{\pi \{ \Delta \}}{\pi \{ (\Delta \supset a) \supset a \}} \\
\text{s} & \quad \frac{\pi \{ ((A \supset B) \supset C) \supset D \}}{\pi \{ A \supset (B \supset C) \supset D \}} \\
\text{c} & \quad \frac{\pi \{ A \supset A \supset B \}}{\pi \{ A \supset B \}} \\
\text{w} & \quad \frac{\pi \{ B \}}{\pi \{ A \supset B \}}
\end{align*}
\]

\[
\begin{align*}
\text{ai} & \quad \frac{\eta \{ (\Delta \supset a) \supset a \}}{\eta \{ \Delta \}} \\
\text{s} & \quad \frac{\eta \{ A \supset (B \supset C) \supset D \}}{\eta \{ ((A \supset B) \supset C) \supset D \}} \\
\text{c} & \quad \frac{\eta \{ A \supset B \}}{\eta \{ A \supset A \supset B \}} \\
\text{w} & \quad \frac{\eta \{ A \supset B \}}{\eta \{ B \}}
\end{align*}
\]
In order to prove cut elimination we must now prove:

$$\Gamma_{\text{SJS}} \quad \rightarrow \quad \Gamma_{\text{JS}}$$

$$A \quad \rightarrow \quad A$$

because cuts must be made atomic to be eliminated, so that at least instances of the dual switch must be eliminated. This fails for open derivations, but we can do:

$$A \quad \rightarrow \quad C$$

$$\begin{array}{c}
A \\
\parallel_{\text{SJS}} \\
B
\end{array} \quad \rightarrow \quad \begin{array}{c}
\parallel_{\text{JS}^\uparrow} \\
C \\
\parallel_{\text{JS}^\downarrow} \\
B
\end{array}$$

Note that this requires to enrich the notion of flow-graph (see paper).
The problem is to find a measure, so we decompose the problem:

1. permute the contractions out of the way
2. permute the weakenings out of the way
3. normalise derivations of the core fragment

Where the core fragment of SJS is composed of the identity and switch rules and their duals. Of course, the complicated part is to permute contractions.
The permutations of contractions are complex, for example:

\[
\begin{align*}
&\pi \left\{ ((A \supset B) \supset C) \supset D \right\} \\
&\frac{c \uparrow \pi \left\{ A \supset (B \supset C) \supset D \right\}}{\pi \left\{ A \supset (B \supset B \supset C) \supset D \right\}} \\
&s \downarrow \frac{c \uparrow \pi \left\{ ((A \supset B) \supset C) \supset D \right\}}{\pi \left\{ (A \supset B) \supset (A \supset B) \supset C \supset D \right\}} \\
&\frac{s \downarrow \pi \left\{ (A \supset A) \supset (B \supset B \supset C) \supset D \right\}}{\pi \left\{ A \supset (B \supset B \supset C) \supset D \right\}}
\end{align*}
\]

introduces a new contraction in the other direction, but there is an argument why this works:

Moving contractions out is terminating because if there is a cycle in the flow-graph of any SJS derivation, there is one in the linear fragment.

Symmetric Normalisation for Intuitionistic Logic
The resulting procedure:

- uses only **local** rewrite rules,
- generalises **cut elimination**,
- implies an **interpolation** result,
- completely **reorganises** the derivation.
Decompositions
The procedure really does:

\[
\begin{array}{c}
A \\
\parallel \text{SJS} \\
B
\end{array} \rightarrow
\begin{array}{c}
A \\
\parallel c\uparrow \\
A_1 \\
\parallel w\uparrow \\
A_2 \\
\parallel \text{core}\uparrow \\
C \\
\parallel \text{core}\downarrow \\
B_2 \\
\parallel w\downarrow \\
B_1 \\
\parallel c\downarrow \\
B
\end{array} \rightarrow
\begin{array}{c}
A \\
\parallel c\uparrow \\
A_1 \\
\parallel w\uparrow \\
A_2 \\
\parallel s\uparrow \\
A_3 \\
\parallel i\uparrow \\
C \\
\parallel i\downarrow \\
B_3 \\
\parallel s\downarrow \\
B_2 \\
\parallel w\downarrow \\
B_1 \\
\parallel c\downarrow \\
B
\end{array}
\]
Such decomposition was previously only obtained for classical logic and for MELL: is it possible in general?

Establishing permutation arguments for other systems might allow to obtain decomposition.

Such decomposition results have interesting applications:

- they can simplify proof search
- they could lead to canonicity

and they can take other forms: for example, focusing can be seen as a cyclic decomposition.
In summary, open derivations in intuitionistic logic can be **normalised** and even **decomposed** by local permutations, in a generalised form of cut elimination.

**Future work** will involve:

- finding a measure to normalise **without** decomposition
- applying the same **technique** to other systems
- studying **computational interpretations** of these rewrites
- **simplifying** the toolkit needed to obtain this result