Analytic Calculi for Substructural Logics: Theory and Applications

Agata Ciabattoni

Vienna University of Technology
agata@logic.at
Substructural logics

- include
  - intuitionistic logic,
  - intermediate logics,
  - relevance logics,
  - linear logic,
  - fuzzy logics,
  - ...

- lack the properties expressed by sequent calculus structural rules
- useful for reasoning about natural language, vagueness, resources, dynamic data structures, algebraic varieties, concurrency ...
This Talk

Theory

- Systematic and automated introduction of sequent and hypersequent calculi

with Kazushige Terui & Nikolaos Galatos

This Talk

Theory

- Systematic and automated introduction of sequent and hypersequent calculi
  with Kazushige Terui & Nikolaos Galatos

Applications

- Extraction of concurrent $\lambda$-calculi
  with Federico Aschieri & Francesco A. Genco
  (LICS 2017, Submitted 2018)

- From hypersequent calculi to natural deduction systems
  with Francesco A. Genco
  (TOCL 2018)
Substructural Logics

Substructural logics

Defined as

- **axiomatic** extensions of Full Lambek calculus $FL$
- **subvarieties** of (pointed) residuated lattices $RL$

Algebraization

For any set $A \cup \{A, B\}$ of formulas,

$$A \vdash_{FL+B} A \text{ iff } \varepsilon[A] \models_{RL+\varepsilon(B)} \varepsilon(A)$$

where $\varepsilon(-)$ is the equation corresponding to $-$.  

Example: Gödel logic

obtained by adding

$$(\alpha \rightarrow \beta) \lor (\beta \rightarrow \alpha)$$

to intuitionistic logic ($FL$ + exchange, weakening and contraction)

or equivalently

$$1 \leq (x \rightarrow y) \lor (y \rightarrow x)$$
Substructural Logics

Substructural logics

Defined as

- axiomatic extensions of Full Lambek calculus \( \text{FL} \)
- subvarieties of (pointed) residuated lattices \( \text{RL} \)

Example: Gödel logic

obtained by adding

\[ (\alpha \to \beta) \lor (\beta \to \alpha) \]

to intuitionistic logic \((\text{FL} + \text{exchange, weakening and contraction})\)
or equivalently

\[ 1 \leq (x \to y) \lor (y \to x) \]

to Heyting algebras \((\text{RL} + \text{commutativity, integrality and idempotency})\)
Why analytic calculi?

Substructural logics

Defined as

- *axiomatic* extensions of Full Lambek calculus $\textbf{FL}$
- *subvarieties* of (pointed) residuated lattices $\textbf{RL}$

Their applicability/usefulness strongly depends on the availability of

Analytic calculi

- useful for establishing various properties of logics
- key for developing automated reasoning methods
Sequent Calculus

Sequents (Gentzen 1934)

\[ A_1, \ldots, A_n \Rightarrow B_1, \ldots, B_m \]

Axioms: E.g. \( A \Rightarrow A, \quad \bot \Rightarrow A \)

Rules:

- Structural
  E.g.
  \[
  \frac{\Gamma, B, A \Rightarrow \Pi}{\Gamma, A, B \Rightarrow \Pi} (e, l) \quad \frac{\Gamma, A, A \Rightarrow \Pi}{\Gamma, A \Rightarrow \Pi} (c, l) \quad \frac{\Gamma \Rightarrow \Pi}{\Gamma, A \Rightarrow \Pi} (w, l)
  \]

- Logical (left and right)
- Cut

\[
\frac{\Gamma \Rightarrow A \quad \Sigma, A \Rightarrow \Pi}{\Gamma, \Sigma \Rightarrow \Pi} \quad \text{Cut}
\]
# Sequent Calculus

## Sequents (Gentzen 1934)

\[ A_1, \ldots, A_n \Rightarrow B \]

### Axioms:
- E.g. \( A \Rightarrow A, \quad \bot \Rightarrow A \)

### Rules:
- **Structural**
  - E.g.
    \[
    \frac{\Gamma, B, A \Rightarrow \Pi}{\Gamma, A, B \Rightarrow \Pi} \quad (e, l) \quad \frac{\Gamma, A, A \Rightarrow \Pi}{\Gamma, A \Rightarrow \Pi} \quad (c, l) \quad \frac{\Gamma \Rightarrow \Pi}{\Gamma, A \Rightarrow \Pi} \quad (w, l)
    \]
- **Logical (left and right)**
- **Cut**
  \[
  \frac{\Gamma \Rightarrow A \quad \Sigma, A \Rightarrow \Pi}{\Gamma, \Sigma \Rightarrow \Pi} \quad \text{Cut}
  \]
Sequent Calculus – the cut rule

\[ \frac{\Gamma \Rightarrow A}{\Gamma, \Sigma \Rightarrow \Pi} \]

\[ \frac{A \Rightarrow \Pi}{\Gamma, \Sigma \Rightarrow \Pi} \text{ Cut} \]

- key to prove completeness w.r.t. Hilbert systems

\[ \frac{A}{A \rightarrow B} \frac{A \rightarrow B}{B} \text{ modus ponens} \]

- bad for proof search

Cut-elimination theorem

Each proof using Cut can be transformed into a proof without Cut.
Sequent Calculus – state of the art

😊 Cut-free sequent calculi have been successfully used

- to prove consistency, decidability, interpolation, . . .
- to give syntactic proofs of algebraic properties for which (in particular cases) semantic methods are not known or do not work well

😢 Many useful and interesting logics have no cut-free sequent calculus
Some extensions of the sequent calculus

A large range of generalizations of sequent calculus have been introduced

- hypersequent calculus (Avron, Mints, Pottinger)
- display calculus (Belnap)
- nested sequents (Brünnler, Fitting)
- deep inference (Guglielmi)
- bunched calculi (Dunn, Mints, . . .)
- labelled systems (Gabbay, Negri, Viganó, . . .)
- systems of rules (Negri)
- many placed sequents (TU Vienna)
- ...
Defining analytic calculi: state of the art

The definition of analytic calculi is usually logic-tailored.

**Steps:**

(i) choosing a framework  
(ii) looking for the “right” inference rule(s)  
(iii) proving cut-elimination
Systematic introduction of (hyper)sequent calculi

Chapter I
The base logic: \textbf{FLe}

- FLe \cong \text{commutative Full Lambek calculus}
- FLe \cong \text{intuitionistic logic without weakening and contraction}
- FLe \cong \text{intuitionistic Linear Logic without exponentials}

**Algebraic semantics:**

A \textit{(bounded pointed) commutative residuated lattice} is

\[ P = \langle P, \land, \lor, \otimes, \rightarrow, \top, 0, 1, \bot \rangle \]

1. \( \langle P, \land, \lor \rangle \) is a lattice with \( \top \) greatest and \( \bot \) least
2. \( \langle P, \otimes, 1 \rangle \) is a commutative monoid.
3. For any \( x, y, z \in P \), \( x \otimes y \leq z \iff y \leq x \rightarrow z \)
4. \( 0 \in P \).
Sequent calculus for **commutative FL**

**FLe**

\[
\begin{array}{c}
A, B, \Gamma \Rightarrow \Pi \\
\frac{A \otimes B, \Gamma \Rightarrow \Pi}{\otimes l}
\end{array}
\]

\[
\begin{array}{c}
\Gamma \Rightarrow A \quad \Delta \Rightarrow B
\end{array}
\]

\[
\frac{\Gamma, \Delta \Rightarrow A \otimes B}{\otimes r}
\]

\[
\begin{array}{c}
\Gamma \Rightarrow A \\
B, \Delta \Rightarrow \Pi
\end{array}
\]

\[
\frac{\Gamma, A \rightarrow B, \Delta \Rightarrow \Pi}{\rightarrow l}
\]

\[
\begin{array}{c}
A, \Gamma \Rightarrow B
\end{array}
\]

\[
\frac{\Gamma \Rightarrow A \rightarrow B}{\rightarrow r}
\]

\[
\begin{array}{c}
A, \Gamma \Rightarrow \Pi \\
B, \Gamma \Rightarrow \Pi
\end{array}
\]

\[
\frac{A \lor B, \Gamma \Rightarrow \Pi}{\lor l}
\]

\[
\begin{array}{c}
\Gamma \Rightarrow A_i
\end{array}
\]

\[
\frac{\Gamma \Rightarrow A_1 \lor A_2}{\lor r}
\]

\[
\begin{array}{c}
0 \Rightarrow 0
\end{array}
\]

\[
\frac{A_i, \Gamma \Rightarrow \Pi}{\land l}
\]

\[
\begin{array}{c}
\Gamma \Rightarrow A \\
\Gamma \Rightarrow B
\end{array}
\]

\[
\frac{\Gamma \Rightarrow A \land B}{\land r}
\]

\[
\begin{array}{c}
\bot \Rightarrow \top
\end{array}
\]

\[
\frac{\Gamma \Rightarrow 0}{0r}
\]

\[
\frac{\Gamma \Rightarrow 1}{1r}
\]

\[
\begin{array}{c}
\bot \Rightarrow \Gamma \\
\Gamma \Rightarrow \Pi
\end{array}
\]

\[
\frac{\bot, \Gamma \Rightarrow \Pi}{\bot l}
\]

\[
\begin{array}{c}
1, \Gamma \Rightarrow \Pi
\end{array}
\]

\[
\frac{\Gamma \Rightarrow \Pi}{1 l}
\]
(Commutative) Substructural Logics
defined by adding Hilbert axioms to the sequent calculus \( \text{FLe} \) (or algebraic equations to commutative residuated lattices).

From axioms to rules: example

- **Contraction:** \( \alpha \to \alpha \otimes \alpha \)

  \[
  \frac{A, A, \Gamma \Rightarrow \Pi}{A, \Gamma \Rightarrow \Pi} \quad (c)
  \]

- **Weakening l:** \( \alpha \to 1 \)

  \[
  \frac{\Gamma \Rightarrow \Pi}{\Gamma, A \Rightarrow \Pi} \quad (w, l)
  \]

- **Weakening r:** \( 0 \to \alpha \)

  \[
  \frac{\Gamma \Rightarrow}{\Gamma \Rightarrow A} \quad (w, r)
  \]
(Commutative) Substructural Logics
defined by adding Hilbert axioms to the sequent calculus $\text{FLe}$ (or algebraic
equations to commutative residuated lattices).

From axioms to rules: example

- **Contraction**: $\alpha \rightarrow \alpha \otimes \alpha$
  \[
  \frac{A, A, \Gamma \Rightarrow \Pi}{A, \Gamma \Rightarrow \Pi} \quad (c)
  \]

- **Weakening l**: $\alpha \rightarrow 1$
  \[
  \frac{\Gamma \Rightarrow \Pi}{\Gamma, A \Rightarrow \Pi} \quad (w, l)
  \]

- **Weakening r**: $0 \rightarrow \alpha$
  \[
  \frac{\Gamma \Rightarrow}{\Gamma \Rightarrow A} \quad (w, r)
  \]

Equivalence between rules and axioms

\[
\vdash \text{FLe+}(\text{axiom}) \quad = \quad \vdash \text{FLe+}(\text{rule})
\]

For which axioms can we do it?
Algebraic Proof Theory


- Which Hilbert axioms can be transformed into rules that preserve cut-elimination?
- Which algebraic equations over residuated lattices are preserved by algebraic completions?

A completion of an algebra $\mathbf{A}$ is a complete algebra $\mathbf{B}$ (i.e. it has arbitrary $\lor$ and $\land$) such that $\mathbf{A} \subseteq \mathbf{B}$. 
Classification

Formulas are **classified** according to the

*polarity* of their connectives w.r.t. a calculus (e.g., FLe)

(J.-M. Andreoli, 1992)

- **Positive** polarity: rule introducing the connective on the *left* is invertible

  \[
  \frac{\Gamma, A \Rightarrow \Pi \quad \Gamma, B \Rightarrow \Pi}{\Gamma, A \lor B \Rightarrow \Pi} \lor l
  \]

  E.g. \[
  \frac{\Gamma, A \Rightarrow \Pi \quad \Gamma, B \Rightarrow \Pi}{\Gamma, A \lor B \Rightarrow \Pi} \lor l
  \]

- **Negative** polarity: rule introducing the connective/quantifier on the *right* is invertible

  \[
  \frac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A \rightarrow B} \rightarrow r
  \]

  E.g. \[
  \frac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A \rightarrow B} \rightarrow r
  \]
Substructural Hierarchy

Definition (AC, Galatos and Terui, LICS 2008)

The classes $\mathcal{P}_n, \mathcal{N}_n$ of positive and negative axioms/equations are:

- $\mathcal{P}_0 := \mathcal{N}_0 :=$ atomic formulas
- $\mathcal{P}_{n+1} := \mathcal{N}_n \mid \mathcal{P}_{n+1} \lor \mathcal{P}_{n+1} \mid \mathcal{P}_{n+1} \otimes \mathcal{P}_{n+1} \mid 1 \mid \bot$
- $\mathcal{N}_{n+1} := \mathcal{P}_n \mid \mathcal{P}_{n+1} \rightarrow \mathcal{N}_{n+1} \mid \mathcal{N}_{n+1} \land \mathcal{N}_{n+1} \mid 0 \mid \top$
## Examples

<table>
<thead>
<tr>
<th>Class</th>
<th>Axiom</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{N}_2$</td>
<td>$\alpha \rightarrow 1$, $\bot \rightarrow \alpha$</td>
<td>weakening</td>
</tr>
<tr>
<td></td>
<td>$\alpha \rightarrow \alpha \otimes \alpha$</td>
<td>contraction</td>
</tr>
<tr>
<td></td>
<td>$\alpha \otimes \alpha \rightarrow \alpha$</td>
<td>expansion</td>
</tr>
<tr>
<td></td>
<td>$\otimes \alpha^n \rightarrow \otimes \alpha^m$</td>
<td>knotted axioms</td>
</tr>
<tr>
<td></td>
<td>$\neg (\alpha \land \neg \alpha)$</td>
<td>weak contraction</td>
</tr>
<tr>
<td>$\mathcal{P}_2$</td>
<td>$\alpha \lor \lnot \alpha$</td>
<td>excluded middle</td>
</tr>
<tr>
<td></td>
<td>$(\alpha \rightarrow \beta) \lor (\beta \rightarrow \alpha)$</td>
<td>prelinearity</td>
</tr>
<tr>
<td>$\mathcal{P}_3$</td>
<td>$\lnot \alpha \lor \lnot \lnot \alpha$</td>
<td>weak excluded middle</td>
</tr>
<tr>
<td></td>
<td>$\neg (\alpha \otimes \beta) \lor (\alpha \land \beta \rightarrow \alpha \otimes \beta)$</td>
<td>(wnm)</td>
</tr>
<tr>
<td>$\mathcal{N}_3$</td>
<td>$((\alpha \rightarrow \beta) \rightarrow \beta) \rightarrow ((\beta \rightarrow \alpha) \rightarrow \alpha)$</td>
<td>Łukasiewicz axiom</td>
</tr>
<tr>
<td></td>
<td>$(\alpha \land \beta) \rightarrow \alpha \otimes (\alpha \rightarrow \beta)$</td>
<td>divisibility</td>
</tr>
</tbody>
</table>

The hierarchy collapses to the class $\mathcal{N}_3$ (Jerabek 2016, commutative case)
Algorithm: from axioms to rules

Ingredient 1
The use of the invertible logical rules of FLe

Ingredient 2: Ackermann Lemma
An algebraic equation \( t \leq u \) is equivalent to a quasiequation \( u \leq x \Rightarrow t \leq x \), and also to \( x \leq t \Rightarrow x \leq u \), where \( x \) is a fresh variable not occurring in \( t, u \).

Example: the sequent \( A \Rightarrow B \) is equivalent to

\[
\begin{align*}
\Gamma \Rightarrow A & \quad \Gamma, B \Rightarrow \Delta \\
\Gamma \Rightarrow B & \\
\Gamma, A \Rightarrow \Delta
\end{align*}
\]

\((\Gamma(\Delta) \text{ fresh metavariable for multisets of formulas (at most one))}\)
From Axioms to Sequent Rules

Algorithm to transform axioms/equations up to the class $\mathcal{N}_2$:

- into "good" structural rules in sequent calculus

- into "good" quasiequations

\[ t_1 \leq u_1 \text{ and } \ldots \text{ and } t_m \leq u_m \Rightarrow t_{m+1} \leq u_{m+1} \]
From Axioms to Sequent Rules

Algorithm to transform axioms/equations up to the class $\mathcal{N}_2$:
- into ”good” structural rules in sequent calculus
- into ”good” quasiequations

\[ t_1 \leq u_1 \text{ and } \ldots \text{ and } t_m \leq u_m \Rightarrow t_{m+1} \leq u_{m+1} \]

Dedekind Completion of Rationals
- For any $X \subseteq \mathbb{Q}$,

\[
X^{\triangleright} = \{ y \in \mathbb{Q} : \forall x \in X. x \leq y \} \\
X^{\triangleleft} = \{ y \in \mathbb{Q} : \forall x \in X. y \leq x \}
\]

- $X$ is closed if $X = X^{\triangleright\triangleleft}$
- $\langle \mathbb{Q}, +, \cdot \rangle$ can be embedded into $\langle C(\mathbb{Q}), +, \cdot \rangle$ with

\[ C(\mathbb{Q}) = \{ X \subseteq \mathbb{Q} : X \text{ is closed} \} \]
From Axioms to Sequent Rules

Algorithm to transform axioms/equations up to the class $\mathcal{N}_2$:
- into "good" structural rules in sequent calculus
- into "good" quasiequations

$t_1 \leq u_1$ and...and $t_m \leq u_m \Rightarrow t_{m+1} \leq u_{m+1}$

Beyond $\mathcal{N}_2$?

Ex. $(\alpha \rightarrow \beta) \lor (\beta \rightarrow \alpha)$

Limitative Result

Analytic sequent calculi $\iff$ Dedekind-MacNeille completion

(AC, N. Galatos and K. Terui. APAL 2012)
Hypersequent calculus

Axioms within the class $P_3$ have the form

$$N_2 \lor N_2 \lor \cdots \lor N_2$$

Hypersequents

$$\Gamma_1 \Rightarrow \Pi_1 | \cdots | \Gamma_n \Rightarrow \Pi_n$$

where for all $i = 1, \ldots n$, $\Gamma_i \Rightarrow \Pi_i$ is a sequent.
Hypersequent calculus

Axioms within the class $\mathcal{P}_3$ have the form

$$\mathcal{N}_2 \lor \mathcal{N}_2 \lor \cdots \lor \mathcal{N}_2$$

Hypersequents

$$\Gamma_1 \Rightarrow \Pi_1 \mid \ldots \mid \Gamma_n \Rightarrow \Pi_n$$

where for all $i = 1, \ldots, n$, $\Gamma_i \Rightarrow \Pi_i$ is a sequent.
Hypersequent calculus

It is obtained embedding sequents into *hypersequents*

\[
\Gamma_1 \Rightarrow \Pi_1 \mid \ldots \mid \Gamma_n \Rightarrow \Pi_n
\]

where for all \( i = 1, \ldots, n \), \( \Gamma_i \Rightarrow \Pi_i \) is a sequent.

\[
\begin{align*}
\frac{G \mid \Gamma \Rightarrow A \quad G \mid A, \Delta \Rightarrow \Pi}{G \mid \Gamma, \Delta \Rightarrow \Pi} & \quad \text{Cut} \\
G \mid \Gamma \Rightarrow A & \quad \text{Identity} \\
G \mid \Gamma \Rightarrow A \quad G \mid B, \Delta \Rightarrow \Pi & \quad \rightarrow l \\
\frac{G \mid \Gamma \Rightarrow A \rightarrow B, \Delta \Rightarrow \Pi}{G \mid \Gamma \Rightarrow A \rightarrow B} & \quad \rightarrow r
\end{align*}
\]

and adding suitable rules to manipulate the additional layer of structure.

\[
\begin{align*}
\frac{G \mid \Sigma \Rightarrow B \mid \Gamma \Rightarrow A \mid G'}{G \mid \Gamma \Rightarrow A \mid \Sigma \Rightarrow B \mid G'} & \quad \text{(ee)} \\
\frac{G}{G \mid \Gamma \Rightarrow A} & \quad \text{(ew)} \\
\frac{G \mid \Gamma \Rightarrow A \mid \Gamma \Rightarrow A}{G \mid \Gamma \Rightarrow A} & \quad \text{(ec)}
\end{align*}
\]
Hypersequent calculus

It is obtained embedding sequents into hypersequents

\[ \Gamma_1 \Rightarrow \Pi_1 \mid \ldots \mid \Gamma_n \Rightarrow \Pi_n \]

where for all \( i = 1, \ldots n \), \( \Gamma_i \Rightarrow \Pi_i \) is a sequent.

\[
\begin{align*}
\text{Cut} & \quad \frac{G \mid \Gamma \Rightarrow A \quad G \mid A, \Delta \Rightarrow \Pi}{G \mid \Gamma, \Delta \Rightarrow \Pi} \\
\text{Identity} & \quad \frac{G \mid A \Rightarrow A}{G \mid A \Rightarrow A}
\end{align*}
\]

\[
\begin{align*}
\rightarrow l & \quad \frac{G \mid \Gamma \Rightarrow A \quad G \mid B, \Delta \Rightarrow \Pi}{G \mid \Gamma, A \rightarrow B, \Delta \Rightarrow \Pi} \\
\rightarrow r & \quad \frac{G \mid A, \Gamma \Rightarrow B}{G \mid \Gamma \Rightarrow A \rightarrow B}
\end{align*}
\]

and adding suitable rules to manipulate the additional layer of structure.

\[
\begin{align*}
\text{ee} & \quad \frac{G \mid \Sigma \Rightarrow B \mid \Gamma \Rightarrow A \mid G'}{G \mid \Gamma \Rightarrow A \mid \Sigma \Rightarrow B \mid G'} \\
\text{ew} & \quad \frac{G \mid \Gamma \Rightarrow A \mid \Gamma \Rightarrow A}{G \mid \Gamma \Rightarrow A} \\
\text{ec} & \quad \frac{G \mid \Gamma \Rightarrow A \mid \Gamma \Rightarrow A}{G \mid \Gamma \Rightarrow A}
\end{align*}
\]
Hypersequent calculus: an example

Gödel logic = Intuitionistic logic + (\(\alpha \rightarrow \beta\)) \lor (\(\beta \rightarrow \alpha\))

\[
\frac{G|\Gamma, \Sigma' \Rightarrow \Delta'}{G|\Gamma, \Sigma \Rightarrow \Delta} \quad \frac{G|\Gamma', \Sigma \Rightarrow \Delta}{G|\Gamma, \Sigma \Rightarrow \Delta | \Gamma', \Sigma' \Rightarrow \Delta'}
\]

(com)


\[
\frac{\beta \Rightarrow \beta \quad \alpha \Rightarrow \alpha}{\alpha \Rightarrow \beta | \beta \Rightarrow \alpha}
\]

(com)

\[
\Rightarrow \alpha \rightarrow \beta | \Rightarrow \beta \rightarrow \alpha
\]

(\(\rightarrow, r\), (\(\rightarrow, r\))}

\[
\Rightarrow (\alpha \rightarrow \beta) \lor (\beta \rightarrow \alpha) | \Rightarrow (\alpha \rightarrow \beta) \lor (\beta \rightarrow \alpha)
\]

(\(\lor_i, r\), (\(\lor_i, r\)}

(\(\lor_i, r\), (\(\lor_i, r\))}

(\(\lor_i, r\), (\(\lor_i, r\))}

(\(\lor_i, r\), (\(\lor_i, r\))}

(\(\lor_i, r\), (\(\lor_i, r\))}

(\(\lor_i, r\), (\(\lor_i, r\))}

(\(\lor_i, r\), (\(\lor_i, r\))}

(\(\lor_i, r\), (\(\lor_i, r\))}

(\(\lor_i, r\), (\(\lor_i, r\))}

(\(\lor_i, r\), (\(\lor_i, r\))}

(\(\lor_i, r\), (\(\lor_i, r\))}

(\(\lor_i, r\), (\(\lor_i, r\))}
From Axioms to Hypersequent Rules

Algorithm to transform axioms/equations up to the class $\mathcal{P}_3$:
- into "good" structural rules in hypersequent calculus
- into "good" analytic clauses

\[ t_1 \leq u_1 \text{ and } \ldots \text{and } t_m \leq u_m \Rightarrow t_{m+1} \leq u_{m+1} \text{ or } \ldots \text{or } t_n \leq u_n \]

Limitative Result

Analytic hypersequent calculi $\implies$ hyperDedekind-MacNeille completion

Example

\[ (((\alpha \rightarrow \beta) \rightarrow \beta) \rightarrow ((\beta \rightarrow \alpha) \rightarrow \alpha) \]
From Axioms to Hypersequent Rules

Algorithm to transform axioms/equations up to the class $\mathcal{P}_3$:
- into "good" structural rules in hypersequent calculus
- into "good" analytic clauses

$t_1 \leq u_1$ and \ldots and $t_m \leq u_m \Rightarrow t_{m+1} \leq u_{m+1}$ or \ldots or $t_n \leq u_n$

Limitative Result

Analytic hypersequent calculi $\Rightarrow$ hyperDedekind-MacNeille completion

Example

$$(((\alpha \rightarrow \beta) \rightarrow \beta) \rightarrow ((\beta \rightarrow \alpha) \rightarrow \alpha)$$
From axioms to rules – our tool

https://www.logic.at/tinc/webaxiomcalc

Input: Hilbert axioms

**Output AxiomCalc for: (a -> b) v (b -> a)**

This axiom is in the class: p(2)

Equivalent Analytic Rule:

\[ G|G+2,D+2 \Rightarrow P+2 \quad G|G+2,D+1 \Rightarrow P+1 \]

Get a [paper](https://www.logic.at/tinc/webaxiomcalc) containing an analytic calculus for FLe/HFLe extended with this axiom!
“Applications”

- closure under algebraic completions of large classes of equations
- decidability results
- standard completeness proofs (i.e. completeness of axiomatic systems w.r.t. algebras whose lattice reduct is the real unit interval $[0, 1]$)
- ...

[Image of a cluttered desk]
“We believe that logics with a cut-free hypersequent calculus could serve as bases for parallel \( \lambda \)-calculi.”

(Avron 1991)

\[ \Gamma_1 \Rightarrow \Pi_1 \mid \ldots \mid \Gamma_n \Rightarrow \Pi_n \]
Curry–Howard correspondence

Logic \{ \text{Formulae} \} \quad \leftrightarrow \quad \text{Proofs}

Computation \{ \text{Types} \} \quad \leftrightarrow \quad \text{Programs}

Making proofs analytic $\iff$ evaluating programs
Curry–Howard correspondence

Logic \{ Formulae \rightarrow Proofs \}

Computation \{ Types \rightarrow Programs \}

Making proofs analytic $\iff$ evaluating programs
Curry–Howard correspondence II

- Introduced for Intuitionistic logic and simply typed lambda calculus
- using Natural Deduction systems

\[ t : A \to B \quad u : A \quad tu : B \]

Proof Transformation $\iff \beta$-reduction in $\lambda$-calculus

\[ \frac{[x : A]}{u : B} \quad \frac{\lambda x. u : A \to B}{(\lambda x. u)t : B} \]

\[ \frac{\Gamma_1}{t : A} \quad \frac{\mathcal{P}}{u[t/x] : B} \]
Curry–Howard correspondence II

- Introduced for Intuitionistic logic and simply typed lambda calculus
- using **Natural Deduction** systems

\[
\begin{align*}
\Gamma_1 & \vdash \lambda x.t : A \to B \\
\vdash t : A \to B & \quad \vdash u : A \\
\therefore tu : B & \\
\end{align*}
\]

- Proof Transformation \(\leftrightarrow\) \(\beta\)-reduction in \(\lambda\)-calculus

\[
\begin{align*}
\Gamma_1 \vdash t : A & \\
\vdash u : B & \\
\vdash \lambda x.u : A \to B & \\
\vdash (\lambda x.u)t : B & \\
\vdash u[t/x] : B &
\end{align*}
\]
Different logics, different models of computation

- classical logic (Griffin 1990, Parigot 1997 ..)
- linear logic
- modal logics
- ...

...
Different logics, different models of computation

- classical logic (Griffin 1990, Parigot 1997 ..)
- linear logic
- modal logics
- ...

Intuitionistic Logic  Classical Logic
(~ A → A control operators)

Concurrent λ-calculi
**From Proof Theory to Computation**

- **Hypersequents**
  \( \Gamma_1 \Rightarrow \Delta_1 \mid \ldots \mid \Gamma_n \Rightarrow \Delta_n \)

- **Natural Deduction**
  
- **Curry–Howard for Intermediate Logics**

- **Concurrent \(\lambda\)-calculi**
  
\[ \lambda n \lambda m a(nm) \parallel_a \lambda p \lambda f f(pf(ax)) \]
Chapter II

From Hypersequents to Natural Deduction
Via ... Systems of rules

(S. Negri, JLC 2016)

Defined on labelled sequents (e.g., “$x R y, \Gamma \Rightarrow \Delta, y : A$”) to capture all normal modal logics formalised by Sahlqvist formulae

Sets of rules connected by:

- order constraints
- formula matching constraints

two-levels:

\[
\frac{\Gamma_1 \Rightarrow \Delta_1 \ldots \Gamma_{n1} \Rightarrow \Delta_{n1}}{\Gamma \Rightarrow \Delta} \quad (top_1)
\]

\[
\frac{\Gamma_k \Rightarrow \Delta_k \ldots \Gamma_{nk} \Rightarrow \Delta_{nk}}{\Gamma \Rightarrow \Delta} \quad (top_k)
\]

\[
\frac{\Gamma \Rightarrow \Delta \ldots \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \quad (bottom)
\]
Two-systems of rules: an example

Gödel Logic = $IL + (A \rightarrow B) \lor (B \rightarrow A)$

\[
\begin{align*}
B, \Gamma_1 &\Rightarrow \Delta_1 \quad (com_1) \\
A, \Gamma_1 &\Rightarrow \Delta_1 \\
\vdots \\
\Gamma &\Rightarrow \Delta \quad (com_1) \\
\end{align*}
\]

\[
\begin{align*}
A, \Gamma_2 &\Rightarrow \Delta_2 \quad (com_2) \\
B, \Gamma_2 &\Rightarrow \Delta_2 \\
\vdots \\
\Gamma &\Rightarrow \Delta \quad (ec) \\
\end{align*}
\]
Embedding two-systems and hypersequents

(AC, Genco – TOCL 2018)

- Any hypersequent derivation can be transformed into a derivation using two systems of rules
- Any hypersequent rule can be rewritten as a two systems of rules, and viceversa

\[ \frac{G | \Gamma'_1 \rightarrow \Delta'_1 \quad \ldots \quad G | \Gamma'_k \rightarrow \Delta'_k}{G | \Gamma_1 \rightarrow \Delta_1 \quad \ldots \quad | \Gamma_n \rightarrow \Delta_n} \]

\[ S_1, \ldots, S_n \text{ sets of sequents} \quad \uparrow\downarrow \quad S_1 \cup \cdots \cup S_n = \{ \Gamma'_i \Rightarrow \Delta'_i \}_{1 \leq i \leq k} \]

\[ \frac{S_1}{\Gamma_1 \Rightarrow \Delta_1} \quad \frac{\ldots}{\ldots} \quad \frac{S_n}{\Gamma_n \Rightarrow \Delta_n} \]

\[ \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \]

\[ \Gamma \Rightarrow \Delta \]
From Hypersequents to Natural Deduction

as a corollary of the embedding

Example: Gödel logic
From Hypersequents to Natural Deduction

as a corollary of the embedding

Example: Gödel logic

\[
\begin{align*}
B, \Gamma_1 \Rightarrow \Delta_1 & \quad A, \Gamma_2 \Rightarrow \Delta_2 \quad \text{embedding} \\
A, \Gamma_1 \Rightarrow \Delta_1 & \quad B, \Gamma_2 \Rightarrow \Delta_2
\end{align*}
\]
Example II: Classical Logic

\[
\Gamma_1, A \Rightarrow \Delta_1 \quad \text{embedding} \quad \Gamma \Rightarrow \Delta
\]

\[
\Gamma \Rightarrow \Delta_1 \\
\Gamma_1 \Rightarrow \Delta_1 \\
\vdots \\
\Gamma \Rightarrow \Delta
\]

\[
A \Rightarrow \\
\vdots \\
F \quad F
\]

\[
\neg A \Rightarrow \\
\vdots \\
F
\]
From Hypersequents to Concurrent $\lambda$-calculi

Chapter III
A case study: Gödel logic

(Aschieri, AC and Genco – LICS 2017)

Natural deduction calculus $\text{NG} := \text{NI} +$
The calculus $\lambda_G$

\[ [x^A : A] \]
\[ \vdots \]
\[ u : B \]
\[ \lambda x^A u : A \to B \]
\[ \frac{u : A \quad t : B}{\langle u, t \rangle : A \land B} \]
\[ \frac{u : A \land B}{u \pi_0 : A} \quad \frac{u : A \land B}{u \pi_1 : B} \]
\[ \Gamma \vdash u : \bot \]
\[ \Gamma \vdash \text{efq}_P(u) : P \]

with $P$ atomic, $P \neq \bot$.

\[ \frac{u : A}{au : B} \quad \frac{v : B}{av : A} \]
\[ \vdots \]
\[ w_1 : F \quad \vdots \]
\[ \frac{w_2 : F}{w_1 \parallel a w_2 : F} \]

\[ w_1 \parallel a w_2 \] has the role of $\nu a(W_1 | W_2)$ in $\pi$-calculus.
The calculus $\lambda^G$

\[
\begin{align*}
\Gamma \vdash u : \bot & \quad \text{with } P \text{ atomic, } P \neq \bot. \\
\Gamma \vdash efq_P(u) : P
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash u : A & \\
\Gamma \vdash av : A \\
\vdots
\end{align*}
\]

\[
\begin{align*}
w_1 : F & \\
\vdots
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash a(w_1 \parallel w_2) : F
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash u : A \\
\Gamma \vdash v : B
\end{align*}
\]

\[
\begin{align*}
w_2 : F & \\
\vdots
\end{align*}
\]

\[
\begin{align*}
w_1 \parallel w_2
\end{align*}
\]

\[
\begin{align*}
\nu a(W_1 | W_2)
\end{align*}
\]
Making proofs analytic: Cross reductions & Code Mobility

\[ \lambda \text{input } y \quad \lambda \text{input } z \]

\[ \lambda \text{input } y \quad \lambda \text{input } z \]

\[ \frac{[y]}{u} \quad \frac{[x]}{v} \]

\[ \frac{A}{B} \quad \frac{B}{A} \]

\[ \vdots \quad \vdots \]

\[ \frac{C}{C} \quad \frac{C}{C} \quad a \]
Making proofs analytic: Cross reductions & Code Mobility

\[
\lambda \text{input } y \rightarrow u \\
\lambda \text{input } z \rightarrow v
\]

\[
\begin{array}{c}
\text{\textbf{\[y\]}} \\
\text{\textbf{\[x\]}} \\
\text{\textbf{\[A\]}} \\
\text{\textbf{\[B\]}} \\
\text{\textbf{\[\ldots\]}} \\
\text{\textbf{\[C\]}} \\
\text{\textbf{\[\ldots\]}} \\
\text{\textbf{\[C\]}} \\
\text{\textbf{\[\ldots\]}} \\
\text{\textbf{\[C\]}} \\
\end{array}
\]

\[
\begin{array}{c}
\text{\textbf{\[x\]}} \\
\text{\textbf{\[y\]}} \\
\text{\textbf{\[v\]}} \\
\text{\textbf{\[u\]}} \\
\text{\textbf{\[A\]}} \\
\text{\textbf{\[B\]}} \\
\text{\textbf{\[\ldots\]}} \\
\text{\textbf{\[C\]}} \\
\text{\textbf{\[\ldots\]}} \\
\text{\textbf{\[C\]}} \\
\end{array}
\]
Making proofs analytic: Cross reductions & Code Mobility
Analytic proofs and Computation

Normalization: ideas from

- Embedding of hypersequents into systems of rules
- Hypersequent cut-elimination

1. Transform the term in parallel form

\[ t_1 \parallel a_1 \quad t_2 \parallel a_2 \cdots \parallel a_n \quad t_{n+1} \]

2. By permutations, isolate the redex \( u \parallel a v \), where \( a \) violates the subformula property with maximal complexity

3. Normalize in parallel \( u \) and \( v \)
   (Simply-typed \( \lambda \)-calculus reductions)

4. Apply cross reductions to \( u \parallel a v \)

Every proof term reduces to an analytic proof
Analytic proofs and Computation

1. Transform the term in parallel form

\[ t_1 \parallel a_1 \ t_2 \parallel a_2 \ldots \parallel a_n \ t_{n+1} \]

2. By permutations, isolate the redex \( u \parallel_a v \), where \( a \) violates the subformula property with maximal complexity

3. Normalize in parallel \( u \) and \( v \)
   (Simply-typed \( \lambda \)-calculus reductions)

4. Apply cross reductions to \( u \parallel_a v \)

   Every proof term reduces to an analytic proof
Computing with $\lambda_G$

- more expressive than simply typed $\lambda$ calculus
- enhanced efficiency via code mobility
It’s more powerful than $\lambda$-calculus

Example: **Parallel disjunction**, i.e.

\[
\begin{align*}
O \cup T & \leftrightarrow^* T \\
OT \cup & \leftrightarrow^* T \\
OFF & \leftrightarrow^* F
\end{align*}
\]

Not expressible in $\lambda$-calculus (by Berry’s sequentiality theorem)

In $\lambda_G$

\[
O := \lambda x^{\text{Bool}} \lambda y^{\text{Bool}} (\text{if } x \text{ then } (\lambda z \lambda k z) \text{ else } (\lambda z \lambda k k)) T(\text{ax}) \\
\|_a (\text{if } y \text{ then } (\lambda z \lambda k z) \text{ else } (\lambda z \lambda k k)) T(\text{ay})
\]
Further example: Classical logic

(F. Aschieri, AC and F. A. Genco, Submitted 2018)

Logic $IL + A \lor \neg A$ \quad $\lambda$-calculus $\lambda_{CL}$

\[
\begin{align*}
[a^A : A] & \quad \frac{u : A}{a^-A u : \bot} \\
[\ldots] & \quad [\ldots] \\
\vdots & \quad \vdots \\
\frac{s : F}{s \parallel_a t : F} & \quad (em) \\
\frac{t : F}{t : F}
\end{align*}
\]

Communication schema (sending data)

\[
C[\overline{au}] \parallel_a D \rightarrow D[u/a]
\]
Classical logic and code mobility

\[ IL + A \lor \neg A \]

\[
\begin{align*}
\lambda \text{ input } y & \quad \lambda \text{ input } y \\
\text{ u } & \quad \text{ u } \\
\end{align*}
\]

\[
\begin{array}{c}
\frac{\frac{A}{\bot}}{
\frac{\quad \gamma}{F
\ \ F
}
\frac{\quad a}{F}
}
\end{array}
\]

\[
\begin{array}{c}
\frac{\frac{y}{u}}{
\frac{\quad \gamma}{F
\ \ F
}
\frac{\quad b}{F}
}
\end{array}
\]

\[ C[\overline{a}u] \parallel_a D \rightarrow (C[\overline{b}y] \parallel_a D) \parallel_b D[u^y/b/a] \]
Logic-based concurrent $\lambda$-calculi

How far can we go?

CL  Gödel logic
Logic-based concurrent $\lambda$-calculi

How far can we go?
Logic-based concurrent \( \lambda \)-calculi

For any axiom \( Ax \in \mathcal{P}'_3 \), let \( \lambda_{Ax} \) be the concurrent \( \lambda \)-calculus corresponding to the logic \( \text{IL} + Ax \).

**Our Results (F. Aschieri, AC and F. A. Genco, Submitted 2018)**

- Every proof term in \( \lambda_{Ax} \) reduces to a **normal** proof term satisfying the **subformula property**
- \( \lambda_{Ax} \) is more powerful than the \( \lambda \)-calculus
- \( \lambda_{Ax} \) allows to send open processes

**Example:** for \( Ax := (A_1 \rightarrow A_2) \lor (A_2 \rightarrow A_3) \lor (A_3 \rightarrow A_1) \)
Logic-based concurrent $\lambda$-calculi

For any axiom $Ax \in \mathcal{P}_3'$, let $\lambda_{Ax}$ be the concurrent $\lambda$-calculus corresponding to the logic $\text{IL} + Ax$.

Our Results (F. Aschieri, AC and F. A. Genco, Submitted 2018)

- Every proof term in $\lambda_{Ax}$ reduces to a normal proof term satisfying the subformula property
- $\lambda_{Ax}$ is more powerful than the $\lambda$-calculus
- $\lambda_{Ax}$ allows to send open processes

Example: for $Ax := (A_1 \rightarrow A_2) \lor (A_2 \rightarrow A_3) \lor (A_3 \rightarrow A_1)$
Process-oriented programming

- Graphs as specification for communication topologies
- Encoded as axioms of the shape $\bigvee (A_i \rightarrow \bigwedge B_j) \in P_3$

Example:

![Diagram of process-oriented programming](image)

encoded as

$$(A_1 \rightarrow A_1 \land A_2 \land A_4) \lor (A_2 \rightarrow A_2 \land A_1) \lor (A_3 \rightarrow A_3 \land A_1 \land A_2) \lor (A_4 \rightarrow A_4 \land \bot)$$
Open questions – a selection

- Climbing up the hierarchy

\[ \mathcal{P}_0 \rightarrow \mathcal{P}_1 \rightarrow \mathcal{P}_2 \rightarrow \mathcal{P}_3 \rightarrow \mathcal{P}_4 \rightarrow \cdots \]

\[ \mathcal{N}_0 \rightarrow \mathcal{N}_1 \rightarrow \mathcal{N}_2 \rightarrow \mathcal{N}_3 \rightarrow \mathcal{N}_4 \rightarrow \cdots \]

- Addition of quantifiers
- Logics without contraction
- Strong normalization
- Expressive power of the introduced concurrent $\lambda$-calculi
- Their use (i) to formalize and reason about concurrent systems (ii) as bases for concurrent functional programming languages
- Relationships with other typed process models
Open questions – a selection

- Climbing up the hierarchy

- Addition of quantifiers
- Logics without contraction
- Strong normalization
- Expressive power of the introduced concurrent $\lambda$-calculi
- Their use (i) to formalize and reason about concurrent systems (ii) as bases for concurrent functional programming languages
- Relationships with other typed process models
Permutation Reductions

\(w(u \parallel a v) \mapsto wu \parallel av, \text{ if } a \text{ does not occur free in } w\)

\(\text{efqp}(w_1 \parallel a w_2) \mapsto \text{efqp}(w_1) \parallel a \text{efqp}(w_2)\)

\((u \parallel a v) \pi_i \mapsto u \pi_i \parallel a v \pi_i\)

\(\lambda x^A (u \parallel a v) \mapsto \lambda x^A u \parallel a \lambda x^A v\)

\((u \parallel a v, w) \mapsto (u, w) \parallel a (v, w), \text{ if } a \text{ does not occur free in } w\)

\((w, u \parallel a v) \mapsto (w, u) \parallel a (w, v), \text{ if } a \text{ does not occur free in } w\)

\((u \parallel a v) \parallel_b w \mapsto (u \parallel_b w) \parallel_a (v \parallel_b w), \text{ if the communication complexity of } b \text{ is greater than 0}\)

\(w \parallel_b (u \parallel a v) \mapsto (w \parallel_b u) \parallel_a (w \parallel_b v), \text{ if the communication complexity of } b \text{ is greater than 0}\)

Cross Reductions

Basic Cross Reductions

\(C[a u] \parallel_a D \mapsto D[u/a]\)

where \(\overline{a} : \neg A, a : A, C[a u], D \) are normal simply typed \(\lambda\)-terms and \(C, D \) simple contexts; the sequence of free variables of \(u\) is empty; \(a\) does not occur in \(u\); \(b\) is fresh

Cross Reductions

\(u \parallel_a v \mapsto u, \text{ if } a \text{ does not occur in } u \) and \(u \parallel_a v \mapsto v, \text{ if } a \text{ does not occur in } v\)

\(C[\overline{a} u] \parallel_a D \mapsto (C[\overline{b}(y)] \parallel_a D) \parallel_b D[u^{b/y}/a]\)

where \(\overline{a} : \neg A, a : A, C[a u], D \) are normal simply typed \(\lambda\)-terms and \(C, D\) simple contexts; \(y\) is the (non-empty) sequence of the free variables of \(u\) which are bound in \(C[a u]\); \(B\) is the conjunction of the types of the variables in \(y\); \(a\) is rightmost; \(b\) is fresh and \(\overline{b} : \neg B, b : B\)
Intuitionistic Reductions
\((\lambda x^A u) t \to u[t/x^A]\)

\(\langle u_0, u_1 \rangle \pi_i \to u_i\), for \(i = 0, 1\)

Permutation Reductions
\((u \parallel_a v) w \to uw \parallel_a vw\), if \(a\) does not occur free in \(w\)

\(w(u \parallel_a v) \to wu \parallel_a wv\), if \(a\) does not occur free in \(w\)

\(\text{efq}_P(w_1 \parallel_a w_2) \to \text{efq}_P(w_1) \parallel_a \text{efq}_P(w_2)\)

\((u \parallel_a v) \pi_i \to u \pi_i \parallel_a v \pi_i\)

\(\lambda x^A (u \parallel_a v) \to \lambda x^A u \parallel_a \lambda x^A v\)

\(\langle u \parallel_a v, w \rangle \to \langle u, w \rangle \parallel_a \langle v, w \rangle\), if \(a\) does not occur free in \(w\)

\(\langle w, u \parallel_a v \rangle \to \langle w, u \rangle \parallel_a \langle w, v \rangle\), if \(a\) does not occur free in \(w\)

\((u \parallel_a v) \parallel_b w \to (u \parallel_b w) \parallel_a (v \parallel_b w)\), if the communication complexity of \(b\) is greater than 0

\(w \parallel_b (u \parallel_a v) \to (w \parallel_b u) \parallel_a (w \parallel_b v)\), if the communication complexity of \(b\) is greater than 0

Cross Reductions
\(u \parallel_a v \to u\) if \(a\) does not occur in \(u\)

\(u \parallel_a v \to v\) if \(a\) does not occur in \(v\)

\(C[a^{A\to B} u] \parallel_a D[a^{B\to A} v] \to
\((D[u^{b^{C\to D} (z)/y}] \parallel_a C[a^{A\to B} u]) \parallel_b (C[v^{b^{D\to C} (y)/z}] \parallel_a D[a^{B\to A} v])\)

where \(C[a u], D[a v]\) are normal simply typed \(\lambda\)-terms and \(C, D\) simple contexts; \(y\) is the sequence of the free variables of \(u\) which are bound in \(C[a u]\); \(z\) is the sequence of the free variables of \(v\) which are bound in \(D[a v]\); \(C\) and \(D\) are the conjunctions of the types of the variables in \(z\) and \(y\), respectively; the displayed occurrences of \(a\) are the rightmost both in \(C[a u]\) and in \(D[a v]\); \(b\) is fresh; and the communication complexity of \(a\) is greater than 0