A Subatomic Proof System

Andrea Aler Tubella
(joint work with Alessio Guglielmi)

University of Bath

23rd May 2015

Deep inference web site: `http://alessio.guglielmi.name/res/cos/`

Outline

Self-dual Non-commutative Logical Operator

What is deep inference?

Proof systems with a single rule

The proof system

Soundness and completeness

Generalised cut-elimination procedure

Self-dual Non-commutative Logical Operator

- (Multiplicative) linear logic is derived from a denotational semantics called coherence spaces [2].
- Coherence spaces naturally interpret three connectives [6]:
  - the commutative $\parr$ (par) and $\otimes$ (tensor) operators of linear logic and
  - a self-dual non-commutative commutative operator $\triangleleft$ (seq).

$$\overline{A \triangleleft B} = \bar{A} \triangleleft \bar{B} \qquad\qquad A \triangleleft B \neq B \triangleleft A$$

self-duality of $\triangleleft$ non-commutativity of $\triangleleft$

A multiplicative linear logic with the three operators exists, it is called BV [5] and it cannot be represented in the sequent calculus [8]. However, deep inference provides BV with a proof theory (in particular cut elimination).

Can we exploit self-dual non-commutativity (in deep inference) for classical logic?

What is deep inference?

It's the free composition of proofs via the same connectives as formulae.

If

$$\Phi = \begin{array}{c} A \\ \| \\ B \end{array} \qquad \text{and} \qquad \Psi = \begin{array}{c} C \\ \| \\ D \end{array}$$

are two proofs with, respectively, premisses A and C and conclusions B and D, then

$$(\Phi \wedge \Psi) = \begin{array}{c} (A \wedge C) \\ \| \\ (B \wedge D) \end{array} \qquad \text{and} \qquad [\Phi \vee \Psi] = \begin{array}{c} [A \vee C] \\ \| \\ [B \vee D] \end{array}$$

are valid proofs with, respectively, premisses $(A \wedge C)$ and $[A \vee C]$, and conclusions $(B \wedge D)$ and $[B \vee D]$.

Proof systems with a single rule

- Goal: generating propositional proofs by a single, linear, simple and regular inference rule scheme.
- Idea: consider atoms as self-dual, noncommutative binary logical relations.
  - This means working with an extended language of formulae (atoms inside atoms, whatever that means). However, if we only look for proofs of the usual formulae, we only obtain proofs with the usual (deep inference) proof theory.
- Why: We can use the rule scheme to reason very generally on proof systems.
  - We are able to prove a general cut-elimination theorem that can be applied to systems as varied as propositional classical logic (KS) [1], linear logic (LLS) [7] and BV [5].

The proof system

We use propositional classical logic as an example.

Idea: occurrences of an atom a are interpretations of more primitive expressions involving a noncommutative binary relation denoted by a.

- Formulae A and B in the relation a, in this order, are denoted by $A\ a\ B$.
- Formulae are built over the two units for disjunction and conjunction, respectively 0 and 1.

Example: the following two expressions are SA formulae:

$$\langle 0\ a\ 1 \rangle \vee \langle 1\ a\ 0 \rangle \qquad \langle 0\ b\ 1 \rangle\ a\ \langle 1\ c\ \langle 1\ d\ 0 \rangle \rangle \wedge 0 \wedge [\langle 0\ a\ 0 \rangle \vee \langle 1\ b\ 1 \rangle]$$

We call tame the formulae where atoms do not appear in the scope of other atoms (e.g., left) and wild the others (e.g., right).

The proof system

To interpret our extended language of formulae, we define an interpretation map $\mapsto$ from tame SA formulae to ordinary formulae such that

$$0\ a\ 1 \mapsto a \quad 1\ a\ 0 \mapsto \bar{a}$$
$$1\ a\ 1 \mapsto 1 \quad 0\ a\ 0 \mapsto 0$$

where $\bar{a}$ denotes the negation of a.

Note

- atoms are self dual: $\overline{A\ a\ B} \equiv \bar{A}\ a\ \bar{B}$
- atoms are not commutative: $A\ a\ B \not\equiv B\ a\ A$ whenever $A \not\equiv B$

We easily extend $\mapsto$ to all the tame SA formulae in the natural way. For example: $\langle 0\ a\ 1 \rangle \vee \langle 1\ a\ 0 \rangle \mapsto a \vee \bar{a} \qquad [0 \vee 0]\ a\ [1 \vee 1] \mapsto a$

The proof system

Consider the usual contraction rule for an atom:

$$\frac{a \vee a}{a}$$

We could obtain this rule via $\mapsto$ as follows:

$$\frac{\langle 0\ a\ 1\rangle \vee \langle 0\ a\ 1\rangle}{[0 \vee 0]\ a\ [1 \vee 1]} \mapsto \frac{a \vee a}{a} \qquad \text{and} \qquad \frac{\langle 1\ a\ 0\rangle \vee \langle 1\ a\ 0\rangle}{[1 \vee 1]\ a\ [0 \vee 0]} \mapsto \frac{\bar{a} \vee \bar{a}}{\bar{a}} \quad .$$

We might consider those rules as generated by the linear scheme

$$\frac{\langle A\ a\ C\rangle \vee \langle B\ a\ D\rangle}{[A \vee B]\ a\ [C \vee D]}$$

This scheme is typical of logical rules in deep inference.

The proof system

Two more examples, identity and cut:

$$\frac{[0 \vee 1]\; a\; [1 \vee 0]}{\langle 0\; a\; 1\rangle \vee \langle 1\; a\; 0\rangle} \mapsto \frac{1}{a \vee \bar{a}} \qquad \text{and} \qquad \frac{\langle 0\; a\; 1\rangle \wedge \langle 1\; a\; 0\rangle}{(0 \wedge 1)\; a\; (1 \wedge 0)} \mapsto \frac{a \wedge \bar{a}}{0} \quad .$$

They are generated by the linear schemes:

$$\frac{[A \vee C]\; a\; [B \vee D]}{\langle A\; a\; B\rangle \vee \langle C\; a\; D\rangle} \qquad \text{and} \qquad \frac{\langle A\; a\; C\rangle \wedge \langle B\; a\; D\rangle}{(A \wedge B)\; a\; (C \wedge D)}$$

The proof system

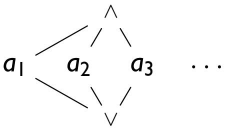
- A subatomic system SA is a deep inference system whose rules are instances of the inference rule scheme

$$\frac{(A \; \alpha \; C) \; \beta \; (B \; \gamma \; D)}{(A \; \delta \; B) \; \epsilon \; (C \; \zeta \; D)}$$

- To represent a particular logical sytem, we specify the sixtuples $\langle \alpha\beta\gamma\delta\epsilon\zeta \rangle$ that generate it.
- A possibility is representing the system by a set of tuples given in terms of a partial ordering of the relations.

The proof system

- Consider the partial order C of logical relations



- $>_C$ corresponds to implication, e.g.: $A \wedge B \Rightarrow A \vee B$ and $0 \wedge 1 \Rightarrow 0\ a\ 1 \Rightarrow 0 \vee 1$.
- On C we define the involution $\bar{\cdot}$ such that $\bar{\vee} = \wedge$ and $\bar{a}_i = a_i$.
- For each of element α of C we define the set $\mathrm{i}(\alpha) = \{\alpha, \bar{\alpha}\}$.

A system for classical logic is given by the tuples in the set

$$Q_C = \{\, \langle \alpha\beta\gamma\beta\alpha\zeta \rangle \mid \alpha \leq_C \zeta, \delta \in \mathrm{i}(\alpha), \gamma \leq_C \beta, \beta \in \mathrm{i}(\zeta) \,\} \setminus \{\langle \vee\wedge\vee\wedge\vee\wedge \rangle\}$$

.

The proof system

- What is a proof in a SA system?

  Example of Classical Logic:
  A proof is a derivation whose premiss is interpreted as 1.

- A proof composed of only tame formulae corresponds to a proof in our usual proof theory.

$$\dfrac{\dfrac{[0 \vee 1]\; a\; [1 \vee 0]}{\langle 0\; a\; 1\rangle \vee \langle 1\; a\; 0\rangle} \wedge \dfrac{[1 \vee 0]\; a\; [0 \vee 1]}{\langle 1\; a\; 0\rangle \vee \langle 0\; a\; 1\rangle}}{\dfrac{\langle 0\; a\; 1\rangle \wedge \langle 1\; a\; 0\rangle}{(0 \wedge 1)\; a\; (1 \wedge 0)} \vee [\langle 1\; a\; 0\rangle \vee \langle 0\; a\; 1\rangle]} \quad \mapsto \quad \dfrac{\dfrac{1}{a \vee \bar{a}} \vee \dfrac{1}{\bar{a} \vee a}}{\dfrac{a \wedge \bar{a}}{0} \vee [\bar{a} \vee a]}$$

Soundness and completeness

We can define the soundness and completeness of a subatomic system by using the interpretation function.

- Soundness: check that each inference rule instance involving tame formulae corresponds to a valid implication between premiss and conclusion.
- Completeness: show that each inference rule (on tame formulae) of a complete system for propositional logic, such as KS, can be represented by one or more rules of SA.

We have sound and complete SA systems for Classical Logic, for BV and for Linear Logic.

Generalised cut-elimination procedure

- The ability to compactly describe all the rules of a system is of much use to reason about it.
- We can give some conditions on the 6-tuples defining a SA system that are sufficient for it to enjoy cut-elimination. (Actually, more than that: admissibility of all the up-rules.)
- Our systems for Classical Logic, Linear Logic, BV verify these conditions.

Generalised cut-elimination procedure

- It is in fact a generalisation of the splitting method.
- In short, it consists on pushing the cut “upwards and outwards”.

Generalised cut-elimination procedure

Theorem (Splitting) *For every proof*

$$\begin{array}{c}\mathsf{t}\\ \| \\ K\{A \wedge B\}\end{array}$$

there are proofs

$$\begin{array}{c}K_A \vee K_B \vee \{\ \}\\ \| \\ K\{\ \}\end{array} \qquad \begin{array}{c}\mathsf{t}\\ \| \\ K_A \vee A\end{array} \qquad \begin{array}{c}\mathsf{t}\\ \| \\ K_B \vee B\end{array}$$

An alike theorem holds for every logic expressed in deep inference so far (including logics that for Gentzen theory are hopeless).

Generalised cut-elimination procedure

Therefore for every cut-free proof $\begin{array}{c}\mathsf{t}\\ \| \\ K\{a \wedge \bar{a}\}\end{array}$

there are cut-free proofs

$$\begin{array}{c}K'\{\bar{a}\} \vee K''\{a\} \vee \{\ \}\\ \| \\ K\{\ \}\end{array} \qquad \begin{array}{c}\mathsf{t}\\ \| \\ K'\{\bar{a}\} \vee a\end{array} \qquad \begin{array}{c}\mathsf{t}\\ \| \\ K''\{a\} \vee \bar{a}\end{array}$$

and so we can build
$$\mathsf{i}{\downarrow}\,\dfrac{\mathsf{t}}{\boxed{\begin{array}{c}a\\ \| \\ K'\{\bar{a}\}\end{array} \vee \begin{array}{c}\bar{a}\\ \| \\ K''\{a\}\end{array}}} \\ \| \\ K\{\mathsf{f}\}$$

and a cut at the bottom would be admissible.

Generalised cut-elimination procedure

$$\dfrac{\dfrac{[0\vee 1]\,a\,[1\vee 0]}{\langle 0\,a\,1\rangle \vee \langle 1\,a\,0\rangle} \wedge \dfrac{[1\vee 0]\,a\,[0\vee 1]}{\langle 1\,a\,0\rangle\vee\langle 0\,a\,1\rangle}}{\boxed{i\uparrow\,\dfrac{\langle 0\,a\,1\rangle\wedge\langle 1\,a\,0\rangle}{(0\wedge 1)\,a\,(1\wedge 0)}} \vee [\langle 1\,a\,0\rangle\vee\langle 0\,a\,1\rangle]} \quad\mapsto\quad \dfrac{\dfrac{1}{a\vee\bar a}\vee\dfrac{1}{\bar a\vee a}}{\boxed{ai\uparrow\,\dfrac{a\wedge\bar a}{0}}\vee[\bar a\vee a]}$$

$$\dfrac{\dfrac{([0\vee 1]\wedge[1\vee 0])}{(0\wedge 1)\vee[1\vee 0]}\; a\; \dfrac{([1\vee 0]\wedge[0\vee 1])}{(1\wedge 0)\vee[0\vee 1]}}{\langle(0\wedge 1)\,a\,(1\wedge 0)\rangle\vee\dfrac{[1\vee 0]\,a\,[0\vee 1]}{\langle 1\,a\,0\rangle\vee\langle 0\,a\,1\rangle}} \quad\mapsto\quad \dfrac{1}{\bar a\vee a}$$

References

[1] K. Brünnler and A. F. Tiu.
A local system for classical logic.
In R. Nieuwenhuis and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 2250 of *Lecture Notes in Computer Science*, pages 347–361. Springer-Verlag, 2001.

[2] J.-Y. Girard.
Linear logic.
Theoretical Computer Science, 50:1–102, 1987.

[3] A. Guglielmi.
Subatomic logic, 2002.

[4] A. Guglielmi.
Some news on subatomic logic, 2005.

[5] A. Guglielmi.
A system of interaction and structure.
ACM Transactions on Computational Logic, 8(1):1:1–64, 2007.

[6] C. Retoré.
Pomset logic: A non-commutative extension of classical linear logic.
In P. De Groote and J. R. Hindley, editors, *Typed Lambda Calculi and Applications (TLCA)*, volume 1210 of *Lecture Notes in Computer Science*, pages 300–318. Springer-Verlag, 1997.

[7] L. Straßburger.
A local system for linear logic.
In M. Baaz and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 2514 of *Lecture Notes in Computer Science*, pages 388–402. Springer-Verlag, 2002.

[8] A. Tiu.
A system of interaction and structure II: The need for deep inference.
Logical Methods in Computer Science, 2(2):4:1–24, 2006.