THE COMMUTATIVE/NONCOMMUTATIVE LINEAR LOGIC BV

ALESSIO GUGLIELMI

Abstract. This brief survey contains an informal presentation of the commutative/noncommutative linear logic BV in terms of a naif space-temporal model. BV improves on the ability of linear logic to model spatial structure by satisfactorily capturing temporal structure in programming languages and quantum physics. I provide a guide to the literature on BV and suggest that the only satisfactory treatment for it can be obtained in the proof-theoretic methodology of deep inference.

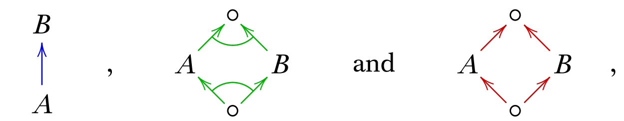
1. An Informal Space-Time Model for BV

In this section I introduce the logic BV by resorting to an intuitive 'space-temporal' model that interprets at the same time its formulae and its proofs. The next section contains some more details on the proof theory developed for and around BV. It turns out that in order to deal with BV we have to change rather deeply our proof-theoretic methods. In fact, despite its simplicity, BV goes beyond what Gentzen's proof theory [10] can handle. On the other hand, the proof-theoretic methodology needed for BV, which we call *deep inference*, benefits all the other logics, so I will survey some of the main results that deep inference makes possible. The formal definition of BV and the development of its basic proof theory can be found in [13].

BV is a very simple object: it is an extension of multiplicative linear logic MLL [12] with the connective $\lhd$, called *seq*. Seq is noncommutative and self-dual, *i.e.*, given two formulae A and B we have that the logical equivalence $A \lhd B = B \lhd A$ only holds if $A = B$; on the other hand the equivalence $\overline{A \lhd B} = \bar{A} \lhd \bar{B}$ holds, where $\bar{\cdot}$ denotes an involutive negation.

MLL, as is well known, contains the two commutative connectives $\parr$ (*par*) and $\otimes$ (*tensor*), which are mutually dual via negation: $\overline{A \parr B} = \bar{A} \otimes \bar{B}$. In MLL negation obeys the two implications $1 \multimap (A \parr \bar{A})$ and $(A \otimes \bar{A}) \multimap \bot$, where 1 and $\bot$ are the units, respectively, for tensor and par. In BV we need to collapse these two units into a single unit, denoted by $\circ$, which also works as a unit for seq: $A \lhd \circ = \circ \lhd A = A$.

We begin to interpret the connectives by looking at the following three diagrams:



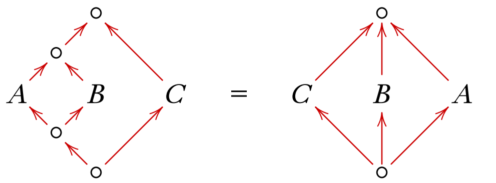
representing $A \lhd B$, $A \parr B$ and $A \otimes B$.

Formulae are statements about events, and the arrows represent a causality relation in time, so that $A \lhd B$ means that all the events that occur in A precede each of the events in B. The two formulae $A \parr B$ and $A \otimes B$ are interpreted as two collections of events A and B that can happen independently in time, and there are points in space-time that precede and follow each of the events in A and B. Those points, which are devoid of events, are represented by small circles and correspond to the unit of BV.

Date: May 8, 2014.

Note that the direction of the arrow is arbitrary: we could have interpreted the flow of time in the opposite direction.

Formula equivalences by associativity, commutativity (only for $\parr$ and $\otimes$) and composition with units are all interpreted in a natural way, so, for example

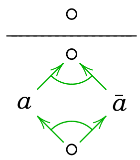


Intuitively, the horizontal dimension is commutative and represents space, while the vertical one is noncommutative and represents time. Formulae, via this bidimensional diagrammatic representation, are statements about a certain structure in space-time. Since (at least so far) we are dealing with a linear logic, we can imagine that an atomic formula stands for an elementary structure at some point in space-time, such as a particle for example.

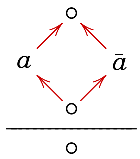
We extend now our interpretation from formulae to proofs, which we consider as chains of implications. We can start from two notable implications that govern negation in BV, *i.e.*, $\circ \multimap (A \parr \bar{A})$ and $(A \otimes \bar{A}) \multimap \circ$. In the case of atomic formulae (which we represent with lowercase letters), the two implications above stand behind the following two elementary inference rules, called $\mathsf{ai}{\downarrow}$ e $\mathsf{ai}{\uparrow}$:

$$\mathsf{ai}{\downarrow}\,\frac{\circ}{a \parr \bar{a}} \qquad \text{and} \qquad \mathsf{ai}{\uparrow}\,\frac{a \otimes \bar{a}}{\circ} \quad ,$$

represented by



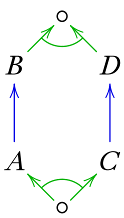
and



.

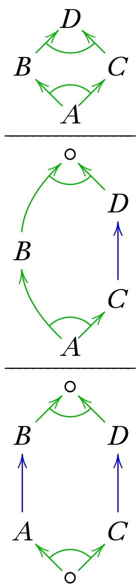
Given the circumstances it is natural to associate to one inference rule the annihilation of two antiparticles and to the other inference rule their creation. There is a perfect top-down symmetry in what we have built so far, therefore the choice is arbitrary. We choose to interpret $\mathsf{ai}{\downarrow}$ as annihilation, or *interaction*, and $\mathsf{ai}{\uparrow}$ as creation. Once this choice is made, the proof theory of BV imposes an interpretation on $\parr$ and $\otimes$ that differentiates them: formulae in a par relation are able to interact while going forward in time, while formulae in a tensor will not interact. If we went backward in time the interpretation would be reversed. Note that from the point of view of proof theory $\mathsf{ai}{\downarrow}$ is an atomic identity (axiom) rule and $\mathsf{ai}{\uparrow}$ is an atomic cut rule.

The interpretation of $A \triangleleft B$ establishes that no space-time point in A can coincide with any space-time point in B, so there is no possible interaction between whatever occupies those two points either going forward or backward in time. Moreover, the temporal structure induced by $\triangleleft$ prevents other interactions. To see this consider for example the formula $F = (A \triangleleft B) \parr (C \triangleleft D)$, represented as



.

In F an interaction event could occur in a space-time point belonging to both B and C, but if this is the case then we cannot have an event between points belonging to both A and D, because F prescribes that whatever happens in D must follow anything in C, and so also anything in A since something in C happens at the same time as something in B. This situation is represented in the following diagrammatic representation of a BV proof, which consists in the application of a single inference rule twice (modulo some unit equations):



Under our space-time interpretation, building a proof means either removing constraints if we read a proof top-to-bottom, or adding constraints if we read it bottom-up. For example, the previous proof can be built bottom-up by starting from its bottom formula and adding the constraint that B and C both happen after A and before D.

Clearly, in order to design a valid proof system one needs to be a lot more careful and precise in formulating its interpretation, or semantics, than I did here. The cited paper [13] develops a proof system for BV under the following assumptions, which are imposed over a precise formulation of the space-time structure mentioned before:

(1) A 'space-time snapshot' should be representable via a formula (and not, for example, via a more general structure such as a circuit).
(2) The interaction mechanism should be as liberal as possible, basically meaning that the removing or adding constraints over space-time, mentioned before, should be the most natural and free that can be conceived.
(3) There should be a mechanism of proof composition equivalent to Gentzen's cut, and there should be a corresponding cut-elimination procedure.

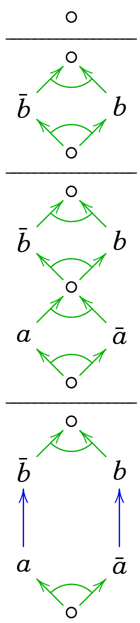
The details of all this go far beyond the scope of this short survey, but I provide some more information in the next section. I conclude this introduction by mentioning two applications, which hopefully help to better understand the interpretation that I offered here and that could serve as motivation for further studying BV.

Arguably, the most primitive constructor for programming languages is sequential composition. In process algebras, such as CCS, we have expressions such as

$$a.\bar{b} \mid \bar{a}.b \quad ,$$

which represents the parallel composition of two processes, $P = a.\bar{b}$ and $Q = \bar{a}.b$, that might communicate over the two channels a and b. Communication (or synchronisation) takes place when two dual atoms representing the same channel are next in the execution order. In the example above the order is right and a communication between a and $\bar{a}$ can take place, followed by a communication between b and $\bar{b}$. In this case we have that the given expression, representing the initial state of the computation, successfully evolves to a state where nothing else has to be done.

Each elementary communication has a very natural interpretation in BV as an instance of ai↓. In fact, a proof corresponding to a successful reduction of $a.\bar{b} \mid \bar{a}.b$ can be represented as follows:



corresponding to $\circ \multimap ((a \lhd \bar{b}) \parr (\bar{a} \lhd b))$.

In the above proof we just apply the principles mentioned before. In particular, we express nondeterministic choice in the execution of a process as nondeterministic choice in imposing constraints to the space-time structure, while building a proof bottom-up. Notice that we are exploiting the self-duality of $\lhd$, which correspond to the self-duality of '.': in fact, we can define $\overline{a.\bar{b}} = \bar{a}.b$ if we want to naturally interpret the two processes P and Q involved in the communication as dual.

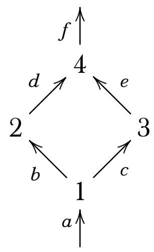
The advantage of BV over MLL should be evident: MLL can naturally model resources and their synchronisation when they belong to a commutative structure (space), but it cannot do so when their natural algebraic structure is noncommutative (time), and especially when it is also self-dual.

There is some ongoing research that uses the previous ideas to obtain refined models of computation involving causality, and not just for process algebras but also for λ-calculi. Some of the early works on the subject are [4, 23] and two recent ones are [27, 28]. I intend to promote and participate in a significant research effort on the subject in the near future mainly because, as I will briefly argue in the next section, other necessary technologies related to the use of deep inference have recently matured enough to justify it.

Similarly to what happens in process algebras, it turns out that the added expressivity of seq is precisely what is needed to overcome the limitations of linear logic when dealing with quantum evolution and entanglement, as argued in the paper [2].

A specific problem to quantum physics is to differentiate between entangled and unentangled particles. At this point it should come as no surprise that the difference can be made by par (for entangled particles) and tensor (for unentangled ones). A direct use of these two constructors for representing the entanglement state leaves the seq connective free for modelling causality.

The following dag of events in a quantum system provides an example. Here the edges represent particles, the nodes represent events, and the arrows establish causality:



.

The four events 1, 2, 3 and 4 can, respectively, be represented in BV as

$$E_1 = \bar{a} \lhd (b \parr c) \quad , \qquad E_2 = \bar{b} \lhd d \quad , \qquad E_3 = \bar{c} \lhd e \qquad \text{and} \qquad E_4 = (\bar{d} \otimes \bar{e}) \lhd f \quad .$$

This representation is not ad hoc, rather it is fully representative of a general way of representing events. By adopting the principles seen so far, it is not too complicated to see that we can obtain a proof for the formula $(a \otimes E_1 \otimes E_2 \otimes E_3 \otimes E_4) \multimap f$, and again this is a general property: the evolution of a quantum system can be faithfully represented in BV under the suggested representation of events.

The difference with what can be obtained with linear logic, so in the absence of seq, is that in order to deal with entailment we have to resort to a rather unnatural encoding that nonetheless is incapable of dealing with correlations that develop dynamically, as the quantum system evolves. A much more thorough explanation can be found in the cited paper [2].

2. The Theory Behind BV

As we have seen the principles behind BV are rather simple and natural, and BV has some interesting applications, but in order to build a proof theory for it we have to radically rethink the traditional proof-theoretic framework that we inherited from Gentzen. Quite simply, it is impossible to obtain cut elimination in Gentzen for the logic BV. Therefore we need a different proof theory, which has been developed in the last 15 years and called *deep inference*. It turns out that the new proof theory has good properties that extend beyond what is necessary for BV, and it benefits all other logics, starting from classical logic and including logics that Gentzen theory cannot properly deal with. This section is a short guide to the core deep-inference literature, with a special emphasis on BV and its extensions. For an almost exhaustive presentation of the literature we refer the reader to the web page `http://alessio.guglielmi.name/res/cos/`.

2.1. Proof Theory of BV The formal semantic structures used to develop BV (in [13]) are called *relation webs*. A relation web is a special kind of graph that generalises the notion of N-free order [26]. The idea is to characterise linear formulae that can arise from any number of commutative and noncommutative relations between atoms, in particular the formulae of BV that we have seen.

Once we have relation webs, we can define an order between logical relations, which in the case of BV is $\otimes > \lhd > \parr$, and stipulate that adding space-temporal constraints means lifting logical relations inside that order. For example, two atoms in

$$
\text{i}\uparrow \frac{A \otimes \bar{A}}{\circ} \qquad \qquad \text{q}\uparrow \frac{(A \lhd C) \otimes (B \lhd D)}{(A \otimes B) \lhd (C \otimes D)}
$$

$$
\text{i}\downarrow \frac{\circ}{A \parr \bar{A}} \qquad \text{s}\frac{(A \parr B) \otimes C}{A \parr (B \otimes C)} \qquad \text{q}\downarrow \frac{(A \parr B) \lhd (C \parr D)}{(A \lhd C) \parr (B \lhd D)}
$$

FIGURE 1. System SBV; system BV is the subsystem consisting of the tree bottom inference rules.

a par can in principle communicate, but we might choose to observe them at different times (so lifting the par to a seq), or to forbid any communication between them (so lifting the par to a tensor). This way we can formally define proofs as upward growing chains of adding constraints. We can then ask the question of what is the most liberal proof system that provides the maximum number of proofs and at the same time admits a cut-elimination procedure.

The answer to this question is the proof system called SBV, in Figure 1, and a few equations about associativity, commutativity and units. The inference rules of SBV require the inspection of the first two levels of the formula tree, as opposed to limiting this, in Gentzen theory, to the root connective. However, this difference is not very significant, considering that in Gentzen one keeps formulae organised in sequents, and we do not do this.

The important difference with Gentzen theory is that we require to compose proofs by the same connectives that we use for formulae. In other words, if

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

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

$$
\begin{array}{c} A \lhd C \\ {\scriptstyle\Phi\lhd\Psi}\Vert \\ B \lhd D \end{array} \quad , \quad \begin{array}{c} A \parr C \\ {\scriptstyle\Phi\parr\Psi}\Vert \\ B \parr D \end{array} \quad \text{and} \quad \begin{array}{c} A \otimes C \\ {\scriptstyle\Phi\otimes\Psi}\Vert \\ B \otimes D \end{array}
$$

are valid proofs with, respectively, premisses $A \lhd C$, $A \parr C$ and $A \otimes C$, and conclusions $B \lhd D$, $B \parr D$ and $B \otimes D$. This is basically the definition of *deep inference*. Technically speaking, there are a few different presentations of deep inference, all of which are equivalent in terms of their main properties. The one that we use here is a formalism called *open deduction*, which has been defined in the paper [15].

For example, a BV proof in open deduction for $\circ \multimap \big((a \lhd \bar{b}) \parr (\bar{a} \lhd b)\big)$ (seen in the previous section) is

$$
\text{q}\downarrow \frac{\boxed{\text{i}\downarrow \frac{\circ}{a \parr \bar{a}}} \lhd \boxed{\text{i}\downarrow \frac{\circ}{b \parr \bar{b}}}}{(a \lhd \bar{b}) \parr (\bar{a} \lhd b)} \quad .
$$

Cut elimination in deep inference is not much different, in principle, from cut elimination in Gentzen theory. At first sight it might seem that the subformula property does not hold in deep inference, but the issue is subtle and it turns out that the differences are surprisingly small. A discussion on this goes beyond the scope of this paper, so I refer the reader to [7].

In practice, however, the more liberal proof composition mechanism of deep inference completely invalidates the techniques (and the intuition) behind cut elimination procedures in Gentzen systems. Much of the effort of these 15 years of research on deep inference went into recovering a normalisation theory. In the case of BV cut elimination consists in proving that the two 'up' rules of SBV, *viz.* i↑ and q↑, are admissible. This is done in [13] by a technique called *splitting* and that is at present the most general method we know for eliminating cuts in deep inference. The technique relies on the fact that reducing cuts to their atomic form is trivial in deep inference, due to the perfect top-down symmetry of the proof systems. Once a cut is reduced to its atomic form, a global transformation is performed on the proof above it. This exposes the atom occurrences that interact with those of the cut and allows us to make them interact between themselves, so eliminating the need for the cut.

A very natural and obvious question is whether BV could be given a 'normal' Gentzen proof theory. Perhaps surprisingly, the answer is that it is impossible. The proof is in the paper [32] by Alwen Tiu. He invented an infinite class of BV tautologies that progressively bury deep into themselves a 'lock'. Tiu then proves that any proof, in any system, needs to undo the lock in order to free atoms that are essential in the subsequent stages of the proof of the tautology. This means that any Gentzen system, being bound by definition over the formula depth that it can reach, will necessarily be invalidated by infinitely many tautologies in the class. Tiu's construction is remarkable, and I suggest the reader to study it in the dissertation where it first appeared [33], because there one can find diagrams in the same space-temporal metaphor that I used here, and they are more intuitive than the relation webs that are used elsewhere.

Ozan Kahramanoğulları proved that BV is NP-complete by encoding into it the 3-partition problem [25]. This means that, as expected, the expressiveness of BV is rather limited. An interesting question is whether it can be extended so that any computation can be expressed in it. The answer is yes. In the three papers [17, 31, 18] Lutz Straßburger and I study a system called NEL, which is BV augmented with the linear logic modalities '?' and '!'. Contrary to BV, system NEL has been proved adequate by Straßburger by encoding into it two-counter machines, which are a universal model of computation [29]. Via an authentic *tour de force* we have been able to devise a cut-elimination procedure for NEL. On the other hand, a simple non-constructive proof of cut admissibility for NEL can be obtained by the technique developed in [11]. Therefore, given its computational and proof-theoretical adequacy, NEL would be a good basis over which we could build new models of computation, as mentioned in the previous section.

On the semantics side I would like to mention the paper [1], where a categorical model of BV is developed and one can find a discussion of previous approaches and possible ways forward.

Finally, I would like to cite the paper [21], which, even if it has nothing to do with BV, is the first attempt to extend deep inference to more general structures than formulae, in this case circuits. Such an investigation might suggest ways to capture even more expressive 'space-temporal' languages than BV.

2.2. Some Comments on Deep Inference and Complexity Besides its unique ability to support BV, deep inference has several other attractive properties. I would only mention here the main points at a high level of abstraction.

The crucial advance of deep inference over Gentzen proof theory is that proof composition is freely built over the same connectives of the underlying logic. This creates a new (top-down) symmetry for proofs, which is needed by BV because it is needed by seq and its inference rules. In order to see this one needs to read Tiu's previously cited papers, but a useful exercise is to try and express the $\mathsf{q}{\downarrow}$ rule in a Gentzen system.

Anyway, the new symmetry has beneficial consequences for two other mechanisms that are typical of proof systems: the cut rule and the contraction rule. Both rules cannot be reduced to their atomic forms in Gentzen except in some cases via global proof transformations. In the case of cut, most of a typical cut elimination procedure in Gentzen goes into making the cut rule atomic, by lifting it toward the premisses. In deep inference this is not the case: every instance of a cut rule can be reduced into several atomic instances by a local procedure, exactly as can be done in Gentzen to an identity axiom. The lack of symmetry in Gentzen only allows this transformation on the instances of identity, but deep inference is top-down symmetric and the problem disappears. The same is true of contraction: a local transformation involving rules that depend on the top-down symmetry makes it possible to always obtain atomic instances of contractions.

Thanks to this, it turns out that normally, for all logics, we can obtain in deep inference proof systems whose rules are all either linear or atomic. This means that the complexity necessary to verify them is bounded by a constant. In other words, we are able to break proofs into their smallest constituents, and so, finally, we can rearrange with unprecedented freedom all those little inference rule instances. This has beneficial consequences in four areas: proof complexity, the proof theory of modal logics, normalisation theory (including the Curry-Howard correspondence) and semantics.

In the proof complexity of propositional logic in deep inference the most spectacular result so far is that cut elimination can be obtained in quasipolynomial time, instead of exponential time as is the case in Gentzen systems [22, 6, 8, 5]. Moreover, Anupam Das recently proved the first new result on the proof complexity of Gentzen systems obtained by adopting deep-inference techniques: in [9] he significantly reduces the best known bound on the size of monotone proofs of the weak pigeonhole principle. There has also been progress in reducing proof-search nondeterminism [24] and in decoupling proof compression mechanisms such as the cut and Tseitin's extension [30].

The proof theory of modal logics has seen an importance advance thanks to a deep-inference based new formalism called *nested sequents.* I will only cite here Kai Brünnler's habilitation thesis on the subject [3], as it is a comprehensive and well written account of all the necessary techniques. Above all, nested sequents provide analytic proof systems for logics for which Gentzen proof theory cannot do so and they do so in a principled way. The cited web page on deep inference provides pointers to a vast literature on nested sequents.

The normalisation theory of classical logic is a subject that received a lot of benefits from atomicity. The small granularity of inference rules has translated into a topological model of normalisation called *atomic flows.* We found that, amazingly, cut elimination in deep inference can be performed in a way that is completely independent of any logical information present in the proof [16, 14]. We are currently using atomic flows, in a relatively large project, in order to design a new deep-inference formalism that removes almost all known sources of 'syntactic bureaucracy', in an attempt to define a purely geometric semantic of proofs and provide an answer to the problem of identity of proofs. Apart from this and the conceptual advance in the study of normalisation, atomic flows have also inspired an explicit-sharing λ-calculus that achieves fully lazy sharing, so improving on the previous results obtained without deep inference [19, 20].

References

[1] Richard Blute, Prakash Panangaden & Sergey Slavnov (2012): *Deep Inference and Probabilistic Coherence Spaces*. *Applied Categorical Structures* 20, pp. 209–228, doi:10.1007/s10485-010-9241-0. Available at `http://www.cs.mcgill.ca/~prakash/Pubs/seqcats.pdf`.

[2] Richard F. Blute, Alessio Guglielmi, Ivan T. Ivanov, Prakash Panangaden & Lutz Straßburger (2014): *A Logical Basis for Quantum Evolution and Entanglement*. In Claudia Casadio, Bob Coecke, Michael Moortgat & Philip Scott, editors: *Categories and Types in Logic, Language, and Physics*, *Lecture Notes in Computer Science* 8222, Springer-Verlag, pp. 90–107, doi:10.1007/978-3-642-54789-8_6. Available at `http://cs.bath.ac.uk/ag/p/LBQEE.pdf`.

[3] Kai Brünnler (2010): *Nested Sequents*. Available at `http://arxiv.org/pdf/1004.1845`. Habilitation Thesis.

[4] Paola Bruscoli (2002): *A Purely Logical Account of Sequentiality in Proof Search*. In Peter J. Stuckey, editor: *Logic Programming, 18th International Conference (ICLP)*, *Lecture Notes in Computer Science* 2401, Springer-Verlag, pp. 302–316, doi:10.1007/3-540-45619-8_21. Available at `http://cs.bath.ac.uk/pb/bvl/bvl.pdf`.

[5] Paola Bruscoli & Alessio Guglielmi (2009): *On the Proof Complexity of Deep Inference*. *ACM Transactions on Computational Logic* 10(2), pp. 14:1–34, doi:10.1145/1462179.1462186. Available at `http://cs.bath.ac.uk/ag/p/PrComplDI.pdf`.

[6] Paola Bruscoli, Alessio Guglielmi, Tom Gundersen & Michel Parigot (2010): *A Quasipolynomial Cut-Elimination Procedure in Deep Inference via Atomic Flows and Threshold Formulae*. In Edmund M. Clarke & Andrei Voronkov, editors: *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-16)*, *Lecture Notes in Computer Science* 6355, Springer-Verlag, pp. 136–153, doi:10.1007/978-3-642-17511-4_9. Available at `http://cs.bath.ac.uk/ag/p/QPNDI.pdf`.

[7] Anupam Das (2011): *On the Proof Complexity of Cut-Free Bounded Deep Inference*. In Kai Brünnler & George Metcalfe, editors: *Tableaux 2011*, *Lecture Notes in Artificial Intelligence* 6793, Springer-Verlag, pp. 134–148, doi:10.1007/978-3-642-22119-4_12. Available at `http://www.anupamdas.com/items/PrCompII/ProofComplexityBoundedDI.pdf`.

[8] Anupam Das (2012): *Complexity of Deep Inference via Atomic Flows*. In S. Barry Cooper, Anuj Dawar & Benedikt Löwe, editors: *Computability in Europe*, *Lecture Notes in Computer Science* 7318, Springer-Verlag, pp. 139–150, doi:10.1007/978-3-642-30870-3_15. Available at `http://www.anupamdas.com/items/RelComp/RelComp.pdf`.

[9] Anupam Das (2014): *On the Pigeonhole and Related Principles in Deep Inference and Monotone Systems*. In: *Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL) and the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. To appear.

[10] Gerhard Gentzen (1969): *Investigations into Logical Deduction*. In M.E. Szabo, editor: *The Collected Papers of Gerhard Gentzen*, North-Holland, Amsterdam, pp. 68–131.

[11] Stéphane Gimenez & Georg Moser (2013): *The Structure of Interaction*. In Simona Ronchi Della Rocca, editor: *Computer Science Logic (CSL)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 23, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp. 316–331, doi:10.4230/LIPIcs.CSL.2013.316. Available at `http://cl-informatik.uibk.ac.at/users/sgimenez/data/articles/soi.pdf`.

[12] Jean-Yves Girard (1987): *Linear Logic*. *Theoretical Computer Science* 50, pp. 1–102, doi:10.1016/0304-3975(87)90045-4.

[13] Alessio Guglielmi (2007): *A System of Interaction and Structure*. *ACM Transactions on Computational Logic* 8(1), pp. 1:1–64, doi:10.1145/1182613.1182614. Available at `http://cs.bath.ac.uk/ag/p/SystIntStr.pdf`.

[14] Alessio Guglielmi & Tom Gundersen (2008): *Normalisation Control in Deep Inference via Atomic Flows*. *Logical Methods in Computer Science* 4(1), pp. 9:1–36, doi:10.2168/LMCS-4(1:9)2008. Available at `http://arxiv.org/pdf/0709.1205.pdf`.

[15] Alessio Guglielmi, Tom Gundersen & Michel Parigot (2010): *A Proof Calculus Which Reduces Syntactic Bureaucracy*. In Christopher Lynch, editor: *21st International Conference on Rewriting Techniques and Applications (RTA)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 6, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp. 135–150, doi:10.4230/LIPIcs.RTA.2010.135. Available at `http://drops.dagstuhl.de/opus/volltexte/2010/2649`.

[16] Alessio Guglielmi, Tom Gundersen & Lutz Straßburger (2010): *Breaking Paths in Atomic Flows for Classical Logic*. In Jean-Pierre Jouannaud, editor: *25th Annual IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, pp. 284–293, doi:10.1109/LICS.2010.12. Available at `http://www.lix.polytechnique.fr/~lutz/papers/AFII.pdf`.

[17] Alessio Guglielmi & Lutz Straßburger (2002): *A Non-commutative Extension of MELL*. In Matthias Baaz & Andrei Voronkov, editors: *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, *Lecture Notes in Computer Science* 2514, Springer-Verlag, pp. 231–246, doi:10.1007/3-540-36078-6_16. Available at `http://www.lix.polytechnique.fr/~lutz/papers/NEL.pdf`.

[18] Alessio Guglielmi & Lutz Straßburger (2011): *A System of Interaction and Structure V: The Exponentials and Splitting*. *Mathematical Structures in Computer Science* 21(3), pp. 563–584, doi:10.1017/S096012951100003X. Available at http://www.lix.polytechnique.fr/~lutz/papers/NEL-splitting.pdf.

[19] Tom Gundersen, Willem Heijltjes & Michel Parigot (2013): *Atomic Lambda Calculus: A Typed Lambda-Calculus with Explicit Sharing*. In Orna Kupferman, editor: *28th Annual IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, pp. 311–320, doi:10.1109/LICS.2013.37. Available at http://opus.bath.ac.uk/34527/1/AL.pdf.

[20] Tom Gundersen, Willem Heijltjes & Michel Parigot (2013): *A Proof of Strong Normalisation of the Typed Atomic Lambda-Calculus*. In Ken McMillan, Aart Middeldorp & Andrei Voronkov, editors: *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-19)*, *Lecture Notes in Computer Science* 8312, Springer-Verlag, pp. 340–354, doi:10.1007/978-3-642-45221-5_24. Available at http://www.cs.bath.ac.uk/~wbh22/docs/SN_atomic_lambda.pdf.

[21] Giorgi Japaridze (2008): *Cirquent Calculus Deepened*. *Journal of Logic and Computation* 18(6), pp. 983–1028, doi:10.1093/logcom/exn019. Available at http://arxiv.org/pdf/0709.1308.

[22] Emil Jeřábek (2009): *Proof Complexity of the Cut-Free Calculus of Structures*. *Journal of Logic and Computation* 19(2), pp. 323–339, doi:10.1093/logcom/exn054. Available at http://www.math.cas.cz/~jerabek/papers/cos.pdf.

[23] Ozan Kahramanoğulları (2005): *Towards Planning as Concurrency*. In M.H. Hamza, editor: *Artificial Intelligence and Applications (AIA)*, ACTA Press, pp. 197–202. Available at http://www.wv.inf.tu-dresden.de/~guglielm/ok/aia05.pdf.

[24] Ozan Kahramanoğulları (2006): *Reducing Nondeterminism in the Calculus of Structures*. In Miki Hermann & Andrei Voronkov, editors: *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, *Lecture Notes in Computer Science* 4246, Springer-Verlag, pp. 272–286, doi:10.1007/11916277_19.

[25] Ozan Kahramanoğulları (2007): *System BV Is NP-Complete*. *Annals of Pure and Applied Logic* 152(1–3), pp. 107–121, doi:10.1016/j.apal.2007.11.005.

[26] Rolf H. Möhring (1989): *Computationally Tractable Classes of Ordered Sets*. In I. Rival, editor: *Algorithms and Order*, *NATO ASI* 255, Kluwer Academic Publishers, pp. 105–193, doi:10.1007/978-94-009-2639-4_4.

[27] Luca Roversi (2011): *Linear Lambda Calculus and Deep Inference*. In Luke Ong, editor: *Typed Lambda Calculi and Applications*, *Lecture Notes in Computer Science* 6690, Springer-Verlag, pp. 184–197, doi:10.1007/978-3-642-21691-6_16. Available at http://www.di.unito.it/~rover/RESEARCH/PUBLICATIONS/2011-TLCA/Roversi2011TLCA.pdf.

[28] Luca Roversi (2014): *A Deep Inference System with a Self-Dual Binder Which Is Complete for Linear Lambda Calculus*. *Journal of Logic and Computation*. Available at http://www.di.unito.it/~rover/RESEARCH/PUBLICATIONS/2014-JLC/Roversi2014JLC.pdf. To appear.

[29] Lutz Straßburger (2003): *System NEL Is Undecidable*. In Ruy De Queiroz, Elaine Pimentel & Lucília Figueiredo, editors: *10th Workshop on Logic, Language, Information and Computation (WoLLIC)*, *Electronic Notes in Theoretical Computer Science* 84, Elsevier, pp. 166–177, doi:10.1016/S1571-0661(04)80853-3. Available at http://www.lix.polytechnique.fr/~lutz/papers/NELundec_wollic03.pdf.

[30] Lutz Straßburger (2012): *Extension Without Cut*. *Annals of Pure and Applied Logic* 163(12), pp. 1995–2007, doi:10.1016/j.apal.2012.07.004. Available at http://www.lix.polytechnique.fr/~lutz/papers/psppp.pdf.

[31] Lutz Straßburger & Alessio Guglielmi (2011): *A System of Interaction and Structure IV: The Exponentials and Decomposition*. *ACM Transactions on Computational Logic* 12(4), pp. 23:1–39, doi:10.1145/1970398.1970399. Available at http://arxiv.org/pdf/0903.5259.

[32] Alwen Tiu (2006): *A System of Interaction and Structure II: The Need for Deep Inference*. *Logical Methods in Computer Science* 2(2), pp. 4:1–24, doi:10.2168/LMCS-2(2:4)2006. Available at http://arxiv.org/pdf/cs.LO/0512036.pdf.

[33] Alwen Fernanto Tiu (2001): *Properties of a Logical System in the Calculus of Structures*. Technical Report WV-01-06, Technische Universität Dresden. Available at http://users.cecs.anu.edu.au/~tiu/thesisc.pdf.

UNIVERSITY OF BATH (UK)
HTTP://ALESSIO.GUGLIELMI.NAME