Deep Inference

Alessio Guglielmi

University of Bath
A.Guglielmi@Bath.Ac.UK

1 Introduction

Deep inference could succinctly be described as an extreme form of linear logic [12]. It is a methodology for designing proof formalisms that generalise Gentzen formalisms, *i.e.* the sequent calculus and natural deduction [11]. In a sense, deep inference is obtained by applying some of the main concepts behind linear logic to the formalisms, *i.e.*, to the rules by which proof systems are designed. By doing so, we obtain a *better* proof theory than the traditional one due to Gentzen. In fact, in deep inference we can provide proof systems for more logics, in a more regular and modular way, with smaller proofs, less syntax, less bureaucracy and we have a chance to make substantial progress towards a solution to the century-old problem of the *identity of proofs*. The first manuscript on deep inference appeared in 1999 and the first refereed papers in 2001 [6, 19]. So far, two formalisms have been designed and developed in deep inference: the *calculus of structures* [15] and *open deduction* [17]. A third one, *nested sequents* [5], introduces deep inference features into a more traditional Gentzen formalism.

Essentially, deep inference tries to understand proof composition and proof normalisation (in a very liberal sense including cut elimination [11]) in the most logic-agnostic way. Thanks to it we obtain a deeper understanding of the nature of normalisation. It seems that normalisation is a primitive, simple phenomenon that manifests itself in more or less complicated ways that depend more on the choice of representation for proofs rather than their true mathematical nature. By dropping syntactic constraints, as we do in deep inference compared to Gentzen, we get closer to the semantic nature of proof and proof normalisation.

As I said, the early inspiration for deep inference comes from linear logic. Linear logic, among other ideas, supports the notion that logic has a *geometric* nature, and that a more perspicuous analysis of proofs is possible if we uncover their geometric shape, hidden behind their syntax. We can give technical meaning to this notion by looking for *linearity* in proofs. In the computing world, linearity can be interpreted as a way to deal with *quantity* or *resource*. The significance of linear logic for computer science has stimulated a remarkable amount of research, that continues to these days, and that ranges from the most theoretical investigations in categorical semantics to the implementation of languages and compilers and the verification of software.

Linear logic expresses locality by relying on Gentzen's formalisms. However, these had been developed for classical mathematical logic, for which linearity is not a primitive, natural notion. While attempting to relate *process algebras* (which are foundational models of concurrent computation) to linear logic, I realised that Gentzen's formalisms were inherently inadequate to express the most primitive notion of composition in computer science: *sequential composition*. This is indeed linear, but of a different kind of linearity from that naturally supported by linear logic.

I realised then that the linear logic ideas were to be carried all the way through and that the formalisms themselves had to be 'linearised'. Technically, this turned out to be possible by dropping one of the assumptions that Gentzen implicitly used, namely that the (geometric) shape of proofs is directly related to the shape of formulae that they prove. In deep inference, we do not make this assumption, and we get proofs whose shape is much more liberally determined than in Gentzen's formalisms. As an immediate consequence, we were able to capture process-algebras sequential composition [7], but we soon realised that the new formalism was offering unprecedented opportunities for both a more satisfying general theory of proofs and for more applications in computer science.

2 Proof System(s)

The difference between Gentzen formalisms and deep inference ones is that in deep inference we compose proofs by the same connectives of 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 $B \vee D$. Significantly, while $\Phi \wedge \Psi$ can be represented in Gentzen, $\Phi \vee \Psi$ cannot. That is basically the definition of deep inference and it holds for every language, not just propositional classical logic.

As an example, I will show the standard deep inference system for propositional logic. System SKS is a proof system defined by the following *structural* inference rules (where a and $\bar{a}$ are dual atoms)

$$\begin{array}{ccc} \mathsf{i}\!\downarrow \dfrac{\mathsf{t}}{a \vee \bar{a}} & \mathsf{w}\!\downarrow \dfrac{\mathsf{f}}{a} & \mathsf{c}\!\downarrow \dfrac{a \vee a}{a} \\ \textit{identity} & \textit{weakening} & \textit{contraction} \\ \\ \mathsf{i}\!\uparrow \dfrac{a \wedge \bar{a}}{\mathsf{f}} & \mathsf{w}\!\uparrow \dfrac{a}{\mathsf{t}} & \mathsf{c}\!\uparrow \dfrac{a}{a \wedge a} \\ \textit{cut} & \textit{coweakening} & \textit{cocontraction} \end{array} ,$$

and by the following two *logical* inference rules:

$$\underset{\textit{switch}}{\mathsf{s}\,\frac{A\wedge[B\vee C]}{(A\wedge B)\vee C}} \qquad \underset{\textit{medial}}{\mathsf{m}\,\frac{(A\wedge B)\vee(C\wedge D)}{[A\vee C]\wedge[B\vee D]}} \quad .$$

A *cut-free* derivation is a derivation where $\mathsf{i}{\uparrow}$ is not used, *i.e.*, a derivation in $\mathsf{SKS}\setminus\{\mathsf{i}{\uparrow}\}$. In addition to these rules, there is a rule

$$=\frac{C}{D} \quad ,$$

such that C and D are opposite sides in one of the following equations:

$$\begin{array}{rclcrcl}
A\vee B &=& B\vee A & \qquad & A\vee\mathsf{f} &=& A\\
A\wedge B &=& B\wedge A & & A\wedge\mathsf{t} &=& A\\
[A\vee B]\vee C &=& A\vee[B\vee C] & & \mathsf{t}\vee\mathsf{t} &=& \mathsf{t}\\
(A\wedge B)\wedge C &=& A\wedge(B\wedge C) & & \mathsf{f}\wedge\mathsf{f} &=& \mathsf{f}
\end{array} \quad .$$

We do not always show the instances of rule $=$, and when we do show them, we gather several contiguous instances into one.

For example, this is a valid derivation:

$$\begin{array}{c}[a\vee b]\wedge a\\ \| \\ ([a\vee b]\wedge a)\wedge([a\vee b]\wedge a)\end{array} = \boxed{\mathsf{m}\,\frac{\boxed{\mathsf{c}{\uparrow}\,\frac{a}{a\wedge a}}\vee\boxed{\mathsf{c}{\uparrow}\,\frac{b}{b\wedge b}}}{[a\vee b]\wedge[a\vee b]}}\wedge\boxed{\mathsf{c}{\uparrow}\,\frac{a}{a\wedge a}} \quad .$$

This derivation illustrates a general principle in deep inference: structural rules on generic formulae (in this case a cocontraction) can be replaced by corresponding structural rules on atoms (in this case $\mathsf{c}{\uparrow}$).

It is interesting to note that the inference rules for classical logic, as well as the great majority of rules for any logic, all derive from a common template which has been distilled from the semantics of a purely linear logic in the first deep inference paper [15]. Since this phenomenon is very surprising, especially for structural rules such as weakening and contraction, we believe that we might be dealing with a rather deep aspect of logic and we are currently investigating it (see [13, 14] for an informal exposition of the idea).

3 Proof-Theoretical Properties

Locality and linearity are foundational concepts for deep inference, in the same spirit as they are for linear logic. Going for locality and linearity basically means going for *complexity bounded by a constant.* This last idea introduces geometry into the picture, because bounded complexity leads us to *equivalence modulo* *continuous deformation.* In a few words, the simple and natural definition of deep inference that we have seen above captures these ideas about linearity, locality and geometry, and can consequently be exploited in many ways, and notably:

– to recover a De Morgan premiss-conclusion symmetry that is lost in Gentzen [3];
– to obtain new notions of normalisation in addition to cut elimination [18, 16];
– to shorten analytic proofs by exponential factors compared to Gentzen [8, 10];
– to obtain quasipolynomial-time normalisation for propositional logic [9];
– to express logics that cannot be expressed in Gentzen [32, 5];
– to make the proof theory of a vast range of logics regular and modular [5];
– to get proof systems whose inference rules are local, which is usually impossible in Gentzen [29];
– to inspire a new generation of proof nets and semantics of proofs [30];
– to investigate the nature of cut elimination [16, 20];
– to type optimised versions of the λ-calculus that are not typeable in Gentzen [21, 22];
– to model process algebras [7, 23, 24, 26, 27];
– to model quantum causal evolution [2] …
– … and much more.

One of the open questions is whether deep inference might have a positive influence on the proof-search-as-computation paradigm and possibly on focusing. This subject has been so far almost unexplored, but some preliminary work looks very promising [25].

The above references have been selected among those that provide surveys when possible. There is no time in this tutorial to cover all those aspects, therefore I refer the reader to this web page, which contains up-to-date information about deep inference:

`http://alessio.guglielmi.name/res/cos` .

The core topic of every proof-theoretic investigation, namely normalisation, deserves a special mention. Traditionally, normalisation is at the core of proof theory, and this is of course the same for deep inference. Normalisation in deep inference is not much different, in principle, from normalisation in Gentzen theory. 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. One of the main ideas is in [15] where we show a technique called *splitting*, which at present is the most general method we know for eliminating cuts in deep inference. Splitting is too technical to be described here. The best reference for those who know Gentzen's theory is probably [4], where splitting is applied to classical predicate logic.

On the other hand, we now have techniques that are not as widely applicable but that are of a completely different nature from splitting, which is combinatorial. A surprising, relatively recent result consists in exploiting deep inference's locality to obtain the first purely geometric normalisation procedure, by a topological device that we call *atomic flows* [16, 18]. This means that, at least for classical logic and logics that extend it, cut elimination can be understood as a process that is completely independent from logical information: only the shape of the proof, determined by its structural information (creation, duplication and erasing of atoms) matters. Logical information, such as the connectives in formulae, *do not matter*. This hints at a deeper nature of normalisation than what we thought so far. It seems that normalisation is a primitive, simple phenomenon that manifests itself in more or less complicated ways that depend more on the choice of representation for proofs rather than their true mathematical nature.

4 Pragmatic Properties

I will concentrate here on a crucial aspect of proofs, namely their size. This is interesting in proof complexity, because proof size is intimately connected to the problem of NP vs coNP. It is also interesting for the automated deduction community, because the size of proofs affects the size of the proof search space, and so it has a direct effect on the time it takes to find proofs.

Quantification in deep inference is not different from quantification in the Gentzen theory, or, at least, nothing significantly different has been discovered so far. Therefore we can limit the discussion to the propositional case. The situation can be described in a few words: in [8] we proved that deep inference has an exponential speed-up over Gentzen on analytic proof systems. In particular, one can consider Statman tautologies [28], which only have exponential-size proofs in the cut-free sequent calculus, and show that they have polynomial proofs in cut-free deep inference.

Obviously, at first sight it might seem that the subformula property does not hold in deep inference, and so that the notion of cut free-ness is weaker than in Gentzen. However, the issue is subtle and it turns out that the differences with Gentzen are surprisingly small. As Anupam Das proved in [10], only a very limited amount of deep inference is sufficient to completely capture the exponential speed-up. More precisely, any cut-free deep-inference system that can access at most depth 2 in formulae can polynomially simulate proof systems of unbounded depth, such as the system presented in this tutorial. In other words, the same depth visibility of hypersequents is sufficient to obtain small proofs. This means that for the same impact that hypersequents have on the branching factor in the proof search space, we can obtain much smaller proofs than in Gentzen systems, thanks to the better proof representation in deep inference. I will show an example here, by reasoning on the first three Statman tautologies (see [8, 28] for formal definitions):

$$\begin{aligned} S_1 &= (a \wedge b) \vee \bar{a} \vee \bar{b} \quad , \\ S_2 &= (c \wedge d) \vee \left([\bar{c} \vee \bar{d}] \wedge a \wedge [\bar{c} \vee \bar{d}] \wedge b \right) \vee \bar{a} \vee \bar{b} \quad , \\ S_3 &= (e \wedge f) \vee \left([\bar{e} \vee \bar{f}] \wedge c \wedge [\bar{e} \vee \bar{f}] \wedge d \right) \vee \\ &\quad \left([\bar{e} \vee \bar{f}] \wedge [\bar{c} \vee \bar{d}] \wedge a \wedge [\bar{e} \vee \bar{f}] \wedge [\bar{c} \vee \bar{d}] \wedge b \right) \vee \bar{a} \vee \bar{b} \quad . \end{aligned}$$

It is well known, and the reader will have no difficulty in seeing it, that the size of cut-free sequent proofs of S_n grows exponentially with n. The structural reason is that the external connectives in formulae force repeated duplication of the context. Let us see what happens if we could just access connectives immediately below the external ones.

For S_1 we have a trivial cut-free proof in SKS:

$$\begin{array}{c} \mathsf{t} \\ \Big\Vert\, \mathsf{i}{\downarrow},\mathsf{s} \\ \boxed{(a \wedge b) \vee \bar{a} \vee \bar{b}} \end{array} \quad = \quad \mathsf{s}\,\frac{\boxed{\mathsf{i}{\downarrow}\,\dfrac{\mathsf{t}}{a \vee \bar{a}}} \wedge \boxed{\mathsf{i}{\downarrow}\,\dfrac{\mathsf{t}}{b \vee \bar{b}}}}{\boxed{\mathsf{s}\,\dfrac{[a \vee \bar{a}] \wedge b}{(a \wedge b) \vee \bar{a}}} \vee \bar{b}} \quad .$$

For S_2 we can obtain:

$$\begin{array}{c} \mathsf{t} \\ \Big\Vert\, \mathsf{i}{\downarrow},\mathsf{s} \\ \boxed{\boxed{\begin{array}{c} \boxed{\boxed{\begin{array}{c} \mathsf{t} \\ \Vert\, \mathsf{i}{\downarrow},\mathsf{s} \\ (c \wedge d) \vee \bar{c} \vee \bar{d} \end{array}} \wedge a \wedge \boxed{\begin{array}{c} \mathsf{t} \\ \Vert\, \mathsf{i}{\downarrow},\mathsf{s} \\ (c \wedge d) \vee \bar{c} \vee \bar{d} \end{array}} \wedge b} \\ \Big\Vert\, \mathsf{s} \\ \boxed{\boxed{\begin{array}{c} (c \wedge d) \vee (c \wedge d) \\ \Vert\, \mathsf{c}{\downarrow},\mathsf{m} \\ \boxed{c \wedge d} \end{array}} \vee \left([\bar{c} \vee \bar{d}] \wedge a \wedge [\bar{c} \vee \bar{d}] \wedge b \right)} \end{array}} \vee \bar{a} \vee \bar{b}} \end{array}$$

Here we see how the external atoms c and d are 'brought inside' the tautology and two proofs similar to those for S_1 are performed inside a conjunction inside the external disjunction.

Finally, in Figure 1 we can see a proof of S_3, where the above principle is repeated and clearly gives rise to a sequence of proofs for S_n that grows polynomially over n instead of exponentially.

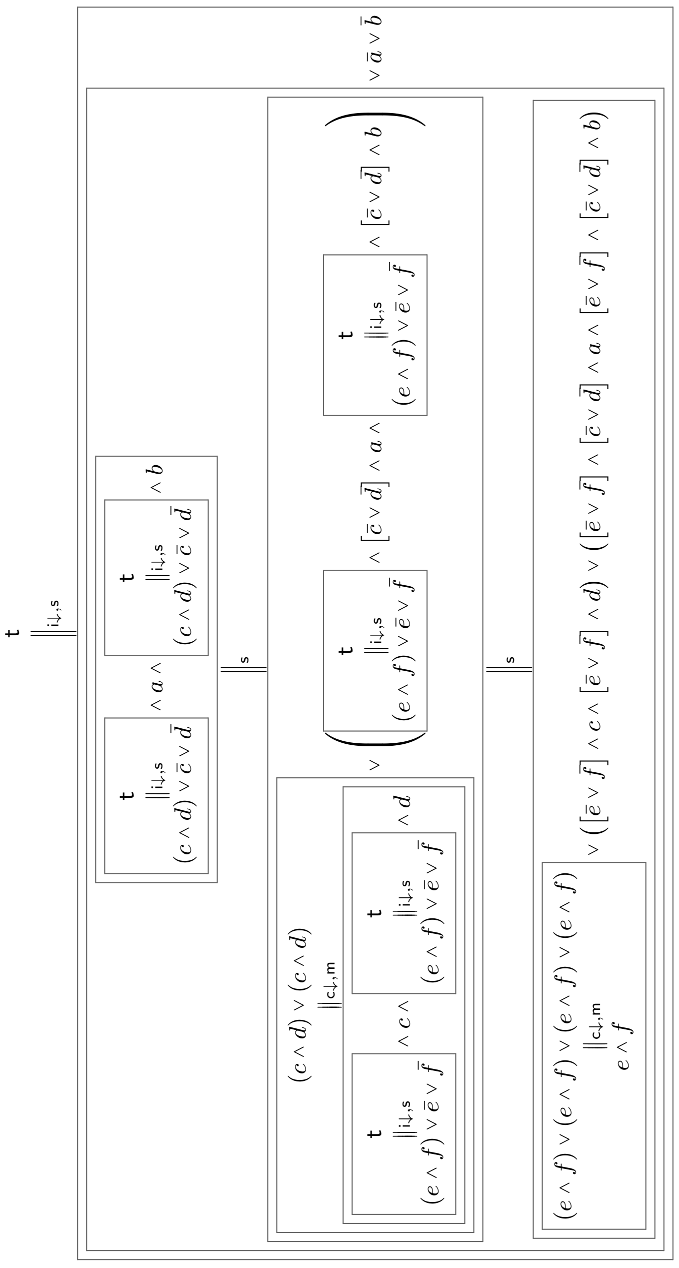

Fig. 1. Cut-free SKS proof of Statman tautology S_3.

5 Trends and Open Problems

The future of deep inference tends towards proof complexity, combinatorics and the study of proofs via algebraic topology. One of the most important open problems that deep inference intends to solve is that of the *identity of proofs* (sometimes called *Hilbert's 24th problem* [31]); this is related to the equally open problem of the *identity of algorithms* [1].

Acknowledgments. This work has been supported by the EPSRC Project *Efficient and Natural Proof Systems* (EP/K018868/1).

References

1. Andreas Blass, Nachum Dershowitz, and Yuri Gurevich. When are two algorithms the same? Technical Report MSR-TR-2008-20, Microsoft Research, 2008.
2. Richard F. Blute, Alessio Guglielmi, Ivan T. Ivanov, Prakash Panangaden, and Lutz Straßburger. A logical basis for quantum evolution and entanglement. In Claudia Casadio, Bob Coecke, Michael Moortgat, and Philip Scott, editors, *Categories and Types in Logic, Language, and Physics*, volume 8222 of *Lecture Notes in Computer Science*, pages 90–107. Springer-Verlag, 2014.
3. Kai Brünnler. *Deep Inference and Symmetry in Classical Proofs.* Logos Verlag, Berlin, 2004.
4. Kai Brünnler. Cut elimination inside a deep inference system for classical predicate logic. *Studia Logica*, 82(1):51–71, 2006.
5. Kai Brünnler. Nested sequents. Habilitation Thesis, 2010.
6. Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. In R. Nieuwenhuis and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 2250 of *Lecture Notes in Computer Science*, pages 347–361. Springer-Verlag, 2001.
7. Paola Bruscoli. A purely logical account of sequentiality in proof search. In Peter J. Stuckey, editor, *Logic Programming, 18th International Conference (ICLP)*, volume 2401 of *Lecture Notes in Computer Science*, pages 302–316. Springer-Verlag, 2002.
8. Paola Bruscoli and Alessio Guglielmi. On the proof complexity of deep inference. *ACM Transactions on Computational Logic*, 10(2):14:1–34, 2009.
9. Paola Bruscoli, Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A quasipolynomial cut-elimination procedure in deep inference via atomic flows and threshold formulae. In Edmund M. Clarke and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-16)*, volume 6355 of *Lecture Notes in Computer Science*, pages 136–153. Springer-Verlag, 2010.
10. Anupam Das. On the proof complexity of cut-free bounded deep inference. In Kai Brünnler and George Metcalfe, editors, *Tableaux 2011*, volume 6793 of *Lecture Notes in Artificial Intelligence*, pages 134–148. Springer-Verlag, 2011.
11. Gerhard Gentzen. Investigations into logical deduction. In M.E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, 1969.
12. Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
13. Alessio Guglielmi. Subatomic logic, 2002.

14. Alessio Guglielmi. Some news on subatomic logic, 2005.
15. Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1:1–64, 2007.
16. Alessio Guglielmi and Tom Gundersen. Normalisation control in deep inference via atomic flows. *Logical Methods in Computer Science*, 4(1):9:1–36, 2008.
17. Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A proof calculus which reduces syntactic bureaucracy. In Christopher Lynch, editor, *21st International Conference on Rewriting Techniques and Applications (RTA)*, volume 6 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 135–150. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2010.
18. Alessio Guglielmi, Tom Gundersen, and Lutz Straßburger. Breaking paths in atomic flows for classical logic. In Jean-Pierre Jouannaud, editor, *25th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 284–293. IEEE, 2010.
19. Alessio Guglielmi and Lutz Straßburger. Non-commutativity and MELL in the calculus of structures. In L. Fribourg, editor, *Computer Science Logic (CSL)*, volume 2142 of *Lecture Notes in Computer Science*, pages 54–68. Springer-Verlag, 2001.
20. Tom Gundersen. *A General View of Normalisation Through Atomic Flows*. PhD thesis, University of Bath, 2009.
21. Tom Gundersen, Willem Heijltjes, and Michel Parigot. Atomic lambda calculus: A typed lambda-calculus with explicit sharing. In Orna Kupferman, editor, *28th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 311–320. IEEE, 2013.
22. Tom Gundersen, Willem Heijltjes, and Michel Parigot. A proof of strong normalisation of the typed atomic lambda-calculus. In Ken McMillan, Aart Middeldorp, and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-19)*, volume 8312 of *Lecture Notes in Computer Science*, pages 340–354. Springer-Verlag, 2013.
23. Ozan Kahramanoğulları. Towards planning as concurrency. In M.H. Hamza, editor, *Artificial Intelligence and Applications (AIA)*, pages 197–202. ACTA Press, 2005.
24. Ozan Kahramanoğulları. On linear logic planning and concurrency. *Information and Computation*, 207(11):1229–1258, 2009.
25. Ozan Kahramanoğulları. Interaction and depth against nondeterminism in proof search. *Logical Methods in Computer Science*, 10(2):5:1–49, 2014.
26. Luca Roversi. Linear lambda calculus and deep inference. In Luke Ong, editor, *Typed Lambda Calculi and Applications*, volume 6690 of *Lecture Notes in Computer Science*, pages 184–197. Springer-Verlag, 2011.
27. Luca Roversi. A deep inference system with a self-dual binder which is complete for linear lambda calculus. *Journal of Logic and Computation*, 2014. To appear.
28. Richard Statman. Bounds for proof-search and speed-up in the predicate calculus. *Annals of Mathematical Logic*, 15:225–287, 1978.
29. Lutz Straßburger. *Linear Logic and Noncommutativity in the Calculus of Structures*. PhD thesis, Technische Universität Dresden, 2003.
30. Lutz Straßburger. From deep inference to proof nets via cut elimination. *Journal of Logic and Computation*, 21(4):589–624, 2011.
31. Rüdiger Thiele. Hilbert's twenty-fourth problem. *American Mathematical Monthly*, 110:1–24, 2003.
32. Alwen Tiu. A system of interaction and structure II: The need for deep inference. *Logical Methods in Computer Science*, 2(2):4:1–24, 2006.