



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Parsifal

*Preuves Automatiques et Raisonnement sur
des SpécIFicAtions Logiques*

Saclay - Île-de-France

Theme : Programs, Verification and Proofs

Activity
R *eport*

2009

Table of contents

1. Team	1
2. Overall Objectives	1
2.1. Main Themes	1
2.2. Highlights of the year	2
3. Scientific Foundations	2
3.1. General Overview	2
3.2. Proof search in Type Theory	3
3.3. A logic for reasoning about logic specifications	3
3.4. Focused proof systems	3
3.5. Deep Inference and Categorical Axiomatizations	4
3.6. Proof Nets and Combinatorial Characterization of Proofs	4
3.7. A Systematic Approach to Cut Elimination	5
4. Application Domains	5
4.1. Automated theorem proving	5
4.2. Mechanized metatheory	5
5. Software	5
5.1. Abella	5
5.2. Tac	6
6. New Results	6
6.1. A logic for reasoning about logic specifications	6
6.2. Two level of logic	6
6.3. New insights into ∇ -quantification	7
6.4. Foundational aspects of focusing proof systems	7
6.5. A completely symmetric approach to proof and refutation	8
6.6. Subexponentials	8
6.7. A framework for proof systems	9
6.8. Modular Systems for Modal Logics	9
6.9. Generic cut-elimination for Substructural Logics	9
6.10. A Kleene Theorem for Forest Languages	10
6.11. Computation with meta-variables	10
6.12. Encoding constrained transition systems	10
6.13. Cut-elimination in classical logic	10
7. Other Grants and Activities	11
7.1. Actions nationales	11
7.1.1. INFER: ANR on the Theory and Application of Deep Inference	11
7.1.2. PSI: ANR on Proof Search control in Interaction with domain-specific methods	11
7.1.3. CPP: ANR on Confidence, Proofs, and Probabilities	11
7.1.4. Panda: ANR on Parallelism and Distribution Analysis	12
7.2. Actions internationales	12
7.2.1. REDO: Redesigning logical syntax	12
7.2.2. Slimmer: an INRIA and NSF funded international team	12
7.2.3. REUSSI: Research Experiences for US Students at INRIA	12
8. Dissemination	12
8.1. Services to the Scientific Community	12
8.1.1. Prizes and distinctions	12
8.1.2. Organization of Conferences and Workshops	12
8.1.3. Editorial activity	13
8.1.4. Participation in program committees	13
8.1.5. Invited talks at Conferences or Workshops	13

8.2. Teaching	13
8.3. Evaluation	14
8.4. Internship supervision	14
9. Bibliography	14

1. Team

Research Scientist

Kaustuv Chaudhuri [CR INRIA-Saclay]
Joëlle Despeyroux [CR INRIA-Sophia]
Stéphane Lengrand [CR CNRS]
Dale Miller [DR INRIA-Saclay, Team Leader]
Lutz Straßburger [CR INRIA-Saclay]

PhD Student

David Baelde [Allocataire École Normale Supérieure, Lyon, defended 9 Dec 2008]
Olivier Delande [Bourse Monge École Polytechnique, defended 15 Oct 2009]
Ivan Gazeau [ANR Blanc CPP, started 1 October 2009]
Nicolas Guenot [Allocataire Ministère de la Recherche, started 1 October 2008]
Vivek Nigam [Allocataire Mobius, defended 18 Sep 2009]
Alexandre Viel [Allocataire École Normale Supérieure, Paris, started 1 Oct 2008]
François Wirion [Bourse INRIA, started 1 Dec 2008]

Post-Doctoral Fellow

Stefan Hetzl [Post Doctorant INRIA, 1 Sep 2008 - 30 Aug 2009]
Andrew Gacek [Post Doctorant INRIA, 1 Oct 2009 - 30 Aug 2010]
Tom Gundersen [Post Doctorant INRIA, 1 Dec 2009 - 30 Nov 2010]

Visiting Scientist

Chuck Liang [Associate Professor, Hofstra University, NY, USA, 8 Jun - 5 Jul 2009]
Gopalan Nadathur [Professor, University of Minnesota, MN, USA, 2 Jun - 1 Jul 2009]
Elaine Pimentel [Associate Professor, Universidade Federal de Minas Gerais, Brazil, 28 Aug - 12 Oct 2009]

Administrative Assistant

Isabelle Biercewicz [until 31 August 2009]
Marie-Jeanne Gaffard [since 1 September 2009]

2. Overall Objectives

2.1. Main Themes

The aim of the Parsifal team is to develop and exploit the theories of proofs and types to support the specification and verification of computer systems. To achieve these goals, the team works on several level.

- The team has expertise in *proof theory* and *type theory* and conducts basic research in these fields: in particular, the team is developing results that help with the automation of deduction and with the formal manipulation and communication of proofs.
- Based on experience with computational systems and theoretical results, the team *designs* new logical principles, new proof systems, and new theorem proving environments.
- Some of these new designs are appropriate for *implementation* and the team works at developing prototype systems to help validate basic research results.
- By using the implemented tools, the team can develop examples of specifications and verification to test the success of the design and to help suggest new logical and proof theoretic principles that need to be developed in order to improve one's ability to specify and verify.

The foundational work of the team focuses on the proof theory of classical, intuitionistic, and linear logics making use, primarily, of sequent calculus and deep inference formalisms. A major challenge for the team is the reasoning about computational specifications that are written in a relational style: this challenge is being addressed with the introduction of some new approaches to dealing with induction, co-induction, and generic judgments. Another important challenge for the team is the development of normal forms of deduction: such normal forms can be used to greatly enhance the automation of search (one only needs to search for normal forms) and for communicating proofs (and proof certificates) for validation.

The principle application areas of concern for the team currently are in functional programming (e.g., λ -calculus), concurrent computation (e.g., π -calculus), interactive computations (e.g., games), and biological systems.

2.2. Highlights of the year

Alexis Saurin's PhD thesis has won the *Prix de thèse de l'École Polytechnique* and the *Prix de thèse ASTI 2009*. His thesis, titled *Une étude logique du contrôle (appliquée à la programmation fonctionnelle et logique)*, was done within the Parsifal team and was advised by Miller.

Dale Miller has been named Editor-in-Chief of the ACM Transactions on Computational Logic.

3. Scientific Foundations

3.1. General Overview

In the specification of computational systems, logics are generally used in one of two approaches. In the *computation-as-model* approach, computations are encoded as mathematical structures, containing such items as nodes, transitions, and state. Logic is used in an external sense to make statements *about* those structures. That is, computations are used as models for logical expressions. Intensional operators, such as the modals of temporal and dynamic logics or the triples of Hoare logic, are often employed to express propositions about the change in state. This use of logic to represent and reason about computation is probably the oldest and most broadly successful use of logic in computation.

The *computation-as-deduction* approach, uses directly pieces of logic's syntax (such as formulas, terms, types, and proofs) as elements of the specified computation. In this much more rarefied setting, there are two rather different approaches to how computation is modeled.

The *proof normalization* approach views the state of a computation as a proof term and the process of computing as normalization (known variously as β -reduction or cut-elimination). Functional programming can be explained using proof-normalization as its theoretical basis [50] and has been used to justify the design of new functional programming languages [22].

The *proof search* approach views the state of a computation as a sequent (a structured collection of formulas) and the process of computing as the process of searching for a proof of a sequent: the changes that take place in sequents capture the dynamics of computation. Logic programming can be explained using proof search as its theoretical basis [54] and has been used to justify the design of new logic programming languages [53].

The divisions proposed above are informal and suggestive: such a classification is helpful in pointing out different sets of concerns represented by these two broad approaches (reductions, confluence, etc, versus unification, backtracking search, etc). Of course, a real advance in computation logic might allow us merge or reorganize this classification.

Although type theory has been essentially designed to fill the gap between these two kinds of approaches, it appears that each system implementing type theory up to now only follows one of the approaches. For example, the Coq system implementing the Calculus of Inductive Constructions (CIC) uses proof normalization while the Twelf system [57], implementing the Edinburgh Logical Framework (LF, a sub-system of CIC), follows the proof search approach (normalization appears in LF, but it is much weaker than in, say, CIC).

The Parsifal team works on both the proof normalization and proof search approaches to the specification of computation.

3.2. Proof search in Type Theory

Cross-fertilizing ideas between the proof search approach and the proof normalization approach, Lengrand has interacted with the TypiCal (INRIA Saclay) and the πr^2 (INRIA Rocquencourt) project-teams.

In proof assistants based on the proof normalization approach, or Type Theory, it is a hard challenge to design and understand their proof search mechanisms. Based on ideas from [49], a major effort has been spent on using concepts from the proof search approach, like *focused proof systems*, in order to rationalize the implemented mechanisms.

By doing so, we have helped improve the Coq system, by impacting the design of the new version of the tool's proof engine. One of these proof search mechanisms, known as *pattern unification*, has again become a hot topic of Coq's design, after Lengrand's use of Coq to specify a particular algorithm has revealed a drastic need for this missing feature.

It also emerged from Lengrand's interaction with these project-teams, that bridging Type Theory with the proof theory developed at Parsifal confirms the need for more extensionality on the functions programmed in Coq. Efforts to add such extensionality are ongoing.

3.3. A logic for reasoning about logic specifications

Coming up with the design of a logic that allows reasoning richly over relational specifications involving bindings in syntax has been a long standing problem, dating from at least the early papers by McDowell and Miller [52] [7], and by Despeyroux, Leleu, Pfenning, and Schürmann [36], [35] [2], [1]. Relational specifications are popular among many designers and implementers of programming languages and computing specification languages. Almost invariably since specifications need to deal with syntax containing variable bindings. Finding a logic appropriate for this domain has gone through many attempts. Pioneer work here includes work by Despeyroux, Leleu, Pfenning, and Schürmann [2], [1], in the Type Theoretic approach. McDowell and Miller [51] also presented a start at such a logic, with a proof-search approach in mind. Later, Tiu and Miller [55], [9] developed the ∇ -quantifier that provided a significant improvement to the expressiveness of logic. Tiu then went on to enrich the possibilities of such a logic as well allowing for more "nominal" effects to be captured [66], [67].

As described in Section 6.1, the team has recently found completely satisfactory designs for a logic for reasoning about logic specifications.

3.4. Focused proof systems

Several team members have continued their efforts to understand and apply focusing proof systems. Since Andreoli's first focused proof system for full linear logic [23], several efforts have attempted to provide focused proof systems for intuitionistic and classical logics. Liang and Miller [14] have provided the LJF and LKF proof systems that appear to be the most general such focusing systems for intuitionistic and classical logics, respectively. In a second work [18], Liang and Miller have also presented a focusing proof system for combinations of classical, intuitionistic, and linear logics that manage to uncover the previous three proof systems as well as allows proofs in these different systems to communicate via the cut inference rule.

Baelde has also focused proof systems found in his PhD thesis [27] to reconstruct and design useful algorithms used within the model checking literature [15].

3.5. Deep Inference and Categorical Axiomatizations

Deep inference [44], [45] is a novel methodology for presenting deductive systems. Unlike traditional formalisms like the sequent calculus, it allows rewriting of formulas deep inside arbitrary contexts. The new freedom for designing inference rules creates a richer proof theory. For example, for systems using deep inference, we have a greater variety of normal forms for proofs than in sequent calculus or natural deduction systems. Another advantage of deep inference systems is the close relationship to categorical proof theory. Due to the deep inference design one can directly read off the morphism from the derivations. There is no need for a counter-intuitive translation.

One reason for using categories in proof theory is to give a precise algebraic meaning to the identity of proofs: two proofs are the same if and only if they give rise to the same morphism in the category. Finding the right axioms for the identity of proofs for classical propositional logic has for long been thought to be impossible, due to “Joyal’s Paradox”. For the same reasons, it was believed for a long time that it is not possible to have proof nets for classical logic. Nonetheless, Lutz Straßburger and François Lamarche provided proof nets for classical logic in [4], and analyzed the category theory behind them in [48]. In [10] and [63], one can find a deeper analysis of the category theoretical axioms for proof identification in classical logic. Particular focus is on the so-called *medial rule* which plays a central role in the deep inference deductive system for classical logic.

The following research problems are investigated by members of the Parsifal team:

- Find deep inference system for richer logics. This is necessary for making the proof theoretic results of deep inference accessible to applications as they are described in the previous sections of this report.
- Investigate the possibility of focusing proofs in deep inference. As described before, focusing is a way to reduce the non-determinism in proof search. However, it is well investigated only for the sequent calculus. In order to apply deep inference in proof search, we need to develop a theory of focusing for deep inference.
- Use the results on deep inference to find new axiomatic description of categories of proofs for various logics. So far, this is well understood only for linear and intuitionistic logics. Already for classical logic there is no common accepted notion of proof category. How logics like LINC can be given a categorical axiomatisation is completely open.

3.6. Proof Nets and Combinatorial Characterization of Proofs

Proof nets are abstract (graph-like) presentations of proofs such that all “trivial rule permutations” are quotiented away. More generally, we investigate combinatoric objects and correctness criteria for studying proofs independently from syntax. Ideally the notion of proof net should be independent from any syntactic formalism. But due to the almost absolute monopoly of the sequent calculus, most notions of proof nets proposed in the past were formulated in terms of their relation to the sequent calculus. Consequently we could observe features like “boxes” and explicit “contraction links”. The latter appeared not only in Girard’s proof nets [42] for linear logic but also in Robinson’s proof nets [60] for classical logic. In this kind of proof nets every link in the net corresponds to a rule application in the sequent calculus.

The concept of deep inference allows to design entirely new kinds of proof nets. Recent work by Lamarche and Straßburger [62] and [5] have extended the theory of proof nets for multiplicative linear logic to multiplicative linear logic with units. This seemingly small step—just adding the units—had for long been an open problem, and the solution was found only by consequently exploiting the new insights coming from deep inference. A proof net no longer just mimics the sequent calculus proof tree, but rather an additional graph structure that is put on top of the formula tree (or sequent forest) of the conclusion. The work on proof nets within the team is focused on the following two directions

- Extend the work of Lamarche and Straßburger to larger fragments of linear logic, containing the additives, the exponentials, and the quantifiers.

- Finding (for classical logic) a notion of proof nets that is deductive, i.e., can effectively be used for doing proof search. An important property of deductive proof nets must be that the correctness can be checked in linear time. For the classical logic proof nets by Lamarche and Straßburger [4] this takes exponential time (in the size of the net). We hope that eventually deductive proof nets will provide a “bureaucracy-free” formalism for proof search.

3.7. A Systematic Approach to Cut Elimination

One of the main problems of proof theory is to prove cut elimination for new logics. Usually, a cut elimination proof is a tedious case analysis, and, in general, it is very fragile and not modular [43]. That means that a minor change in the deductive system makes the cut elimination proof break down, and for every new system one has to start from scratch.

It is therefore an important research task, to find a more systematic approach to cut elimination proofs. That is to say, to find general guidelines that ensure the cut elimination property for large classes of systems, in a similar way as it has been done for display logics [29].

4. Application Domains

4.1. Automated theorem proving

Proving theorems in classical, intuitionistic, and linear logics is an important activity in a number of formal methods and formalized reasoning domains. While this is a well-worn topic (automated theorem proving for classical logic dates back to at least the early 1960’s), the team has been developing many new insights into the structure of proof and to structuring the search for proofs. We are applying at least some of our efforts to the design of new theorem provers, both automatic and interactive.

4.2. Mechanized metatheory

There has been increasing interest in the international community with the use of formal methods to provide proofs of properties of programs and entire programming languages. The example of proof carrying code is one such example. Two more examples for which the team’s efforts should have important applications are the following two challenges.

Tony Hoare’s Grand Challenge titled “Verified Software: Theories, Tools, Experiments” has as a goal the construction of “verifying compilers” to support a vision of a world where programs would only be produced with machine-verified guarantees of adherence to specified behavior. Guarantees could be given in a number of ways: proof certificates being one possibility.

The PoplMark challenge [25] envisions “a world in which mechanically verified software is commonplace: a world in which theorem proving technology is used routinely by both software developers and programming language researchers alike.” The proposers of this challenge go on to say that a “crucial step towards achieving these goals is mechanized reasoning about language metatheory.” There is clearly a strong overlap in the goals of this challenge and those of part of the Parsifal team.

5. Software

5.1. Abella

Participant: Andrew Gacek.

The earliest versions of the Abella theorem prover was written while Gacek was a PhD student at the University of Minnesota. Now that he is a post doc within Parsifal, he has continued to enhance that prover: in particular, he has added some additional static checking for theories (simple type checking), provided modularity for specifications and for theories, and has documented several more example theories. The system is available via the web and several people and groups are known to be using the prover on a regular basis.

5.2. Tac

Participants: David Baelde, Dale Miller, Zachary Snow, Alexandre Viel.

Given the team's expertise with the structure of proofs and techniques for automation, we have taken on the implementation of the TAC prover. This prover, written in OCaml, is currently underdevelopment and has not yet been released. Its architecture is designed to perform focused proof search of rather limited depth and to use only "obvious" induction invariants. A goal of this prover is to completely automate a large number of simple theorems within an inductive and co-inductive setting: proofs of more significant theorems would then be organized as being simple lists of lemmas. A large class of examples, including those from the POPLMark challenge [25], are currently being treated by this prover.

6. New Results

6.1. A logic for reasoning about logic specifications

Participants: David Baelde, Andrew Gacek, Dale Miller.

As described in Section 3.3, there has been a decade-long effort to design a logical framework for reasoning about logic specifications. Finally in 2008 and 2009 team members have reached what appears to be a natural culmination of this development. In particular, David Baelde's PhD [27] and Andrew Gacek's PhD thesis [37] provided rich analysis of how the ∇ -quantification can be related to fixed point definitions and their associated induction and co-induction inference rules. Baelde has concentrated on proving focusing-style results that are critical for proof automation and on a *minimal generic* interpretation of the ∇ -quantifier. Gacek has concentrated on a *nominal generic* interpretation of the ∇ -quantifier. We now understand the difference between these logics: the nominal approach resembles much more closely the approach developed by Pitts [59].

Full proofs of the important meta-theory results of the logic in Gacek's thesis have been submitted for publication [41]. Gacek has also provided an implementation of his logic within the Abella prover that he has worked on as part of his PhD thesis.

We have developed extensive examples of our this new logic: significant examples taken from the π -calculus have been published in [65] and the Abella distribution contains a large number of examples.

6.2. Two level of logic

Participants: Andrew Gacek, Dale Miller.

Relational descriptions have been widely used in formalizing diverse computational notions, including, for example, operational semantics, typing, acceptance by non-deterministic machines, *etc.* Such relational specifications can be faithfully captured by a (restricted) logical theory over relations. Such a *specification logic* can be picked so that it explicitly treats binding in object languages. Once such a logic is fixed, a natural next question is what devices should be used to prove theorems about specifications written in it. Within the team, we have a second logic, called the *reasoning logic*, to reason about provability in the first logic. To be adequate for this purpose, the reasoning logic should be able to completely encode the specification logic, including notions of binding, such as quantifiers within formulas, for eigenvariables within sequents, and for abstractions within terms. To provide a natural treatment of these aspects, the reasoning logic must encode binding structures as well as their associated notions of scope, free/bound variables, and capture-avoiding substitution. Furthermore, the reasoning logic should possess strong mechanisms for constructing proofs by induction and co-induction.

Within the context of the Gacek’s PhD thesis [37] and the submitted paper [40] by Gacek, Miller, and Nadathur, the logic \mathcal{G} was present: this logic represents relations over λ -terms via definitions of atomic judgments, contains inference rules for induction and co-induction, and includes a special quantifier called ∇ and a related generalization of equality over λ -terms called *nominal abstraction*. The interactive theorem prover Abella [39] [37] implements \mathcal{G} and supports this two-level logic approach to reasoning about computation. Gacek and others have now contributed a large number of interesting examples showing the utility of using this two-level approach to reasoning: see the Abella web site for many examples. In particular, the POPLMark challenge problems 1a and 2a [25] have nice, declarative solutions within Abella.

6.3. New insights into ∇ -quantification

Participants: David Baelde, Andrew Gacek, Dale Miller.

The team has been actively extending the scope of effectiveness ∇ -quantification. As Tiu and Miller have shown in [65], the ∇ quantifier (developed in previous years within the team) provides a completely satisfactory treatment of binding structures in the *finite* π -calculus. Moving this quantifier to treat infinite behaviors via induction and co-induction, required new advances in the underlying proof theory of ∇ -quantification.

The team has explored two different approaches to this problem. David Baelde [27], [28] has developed a minimalist generalization of previous work by Miller and Tiu: he has found what seems to be the simplest extension to that earlier work that allows ∇ to interact properly with fixed points and their inference rules (namely, induction and co-induction). His logical approach allows for a rather careful and rigid understanding of scope in the treatment of the meta-theory of logics and computational specifications.

Another angle has been developed as a result of our close international collaborations. Alwen Tiu, now at the Australian National University, has developed a logic, called LG^ω which extends the earlier, “minimal” approach by introducing the structural rules of strengthening and exchange into the context of generic variables. As a result, the behavior of bindings becomes much more like the behavior of names more generally, while still maintaining much of the status as being binders. In combination with our close colleagues at the University of Minnesota, we have extended this work to include a new definitional principle, called *nabla-in-the-head*, that strengthens our ability to declaratively describe the structure of contexts and proof invariants. This new definitional principle was first presented in [38] and examples of it were presented in [39]. Our colleague, Andrew Gacek (a PhD student at the University of Minnesota and former intern with Parsifal) has also built the Abella proof editor that allows for the direct implementation of this new definitional principle. His system is in distribution and has been used by a number of people to develop examples in this logic.

6.4. Foundational aspects of focusing proof systems

Participants: David Baelde, Kaustuv Chaudhuri, Dale Miller, Alexis Saurin.

Since focusing proof systems seem to be behind much of our computational logic framework, the team has spent some energies developing further some foundational aspects of this approach to proof systems.

Chuck Liang and Miller have recently finished the paper [6] in which a comprehensive approach to focusing in intuitionistic and classical logic was developed. In particular, they present a compact sequent calculus LKU for classical logic organized around the concept of *polarization*. Focused sequent calculi for classical, intuitionistic, and multiplicative-additive linear logics are derived as fragments of the host system by varying the sensitivity of specialized structural rules to polarity information. They identify a general set of criteria under which cut elimination holds in such fragments. From cut elimination leads to a unified proof of the completeness of focusing. Furthermore, each sublogic can interact with other fragments through cut. Under certain circumstances, for example, it is possible for a classical lemma to be used in an intuitionistic proof while preserving intuitionistic provability. They also examine the possibility of defining some classical-linear *hybrid logics*.

Given the team's ambitious to automate logics that require induction and co-induction, we have also looked in detail at the proof theory of fixed points. In particular, David Baelde's recent PhD thesis [27] contains a number of important, foundational theorems regarding focusing and fixed points. In particular, he has examined the logic MALL (multiplicative and additive linear logic). To strengthen this decidable logic into a more general logic, Girard added the exponentials, which allowed for modeling unbounded ("infinite") behavior. Baelde considers, however, the addition of fixed points instead and he has developed the proof theory of the resulting logic. We see this logic as being behind much of the work that the team will be doing in the coming few years.

Alexis Saurin's recent PhD [61] also contains a wealth of new material concerning focused proof system. In particular, he provides a new and modular approach to proving the completeness of focused proof systems as well as develops the theme of multifocusing.

A particular outcome of our work on focused proof search is the use of *maximally multifocused proofs* to help provide sequent calculus proofs a canonicity. In particular, Chaudhuri, Miller, and Saurin have shown in [32] that it is possible to show that maximally multifocused sequent proofs can be placed in one-to-one correspondence with more traditional proof net structures for subsets of MALL.

6.5. A completely symmetric approach to proof and refutation

Participants: Olivier Delande, Dale Miller, Alexis Saurin.

A couple of years ago, Miller and Saurin proposed a neutral approach to proof and refutation. The goal was to describe an entirely neutral setting where a step in a "proof search" could be seen as a step in either building a proof of a formula or a proof of its negation. The early work was limited to essentially a simple generalization to additive logic. Delande was able to generalize that work to capture multiplicative connectives as well. His thesis [11] contains two game semantics for multiplicative additive linear logic (MALL): the first is sequential and the second is concurrent. The concurrent game was used to capture full completeness results between MALL (focused) sequent calculus proofs and winning strategies.

The paper [34] provides a detailed description of the sequent approach to both additive and multiplicative linear logic. In that paper, the search for a proof of B or a refutation of B (i.e., a proof of $\neg B$) can be carried out simultaneously: in contrast, the usual approach in automated deduction views proving B or proving $\neg B$ as two, possibly unrelated, activities. Their approach to proof and refutation is described as a two-player game in which each player follows the same rules. A winning strategy translates to a proof of the formula and a counter-winning strategy translates to a refutation of the formula. The game is described for multiplicative and additive linear logic (MALL). A game theoretic treatment of the multiplicative connectives is intricate and involves two important ingredients. First, labeled graph structures are used to represent positions in a game and, second, the game playing must deal with the failure of a given player and with an appropriate resumption of play. This latter ingredient accounts for the fact that neither player might win (that is, neither B nor $\neg B$ might be provable).

6.6. Subexponentials

Participants: Dale Miller, Vivek Nigam.

The inference rules of a logic define a logical connective in a canonical fashion if the following test is passed: assume that there are two copies of a logical connective, say a red and blue copy, and assume that both of these connectives have the same introduction rules. If it is possible to prove that the red and blue versions are equivalent within the extended proof system, then we say that the connective is defined canonically. In linear logic, all connectives are canonical in this sense except for the exponentials ($!$, $?$). That is, it is possible to have many exponentials and they do not need to all support weakening and contraction but some subsets of these structural rules. Since they do not need to provide all structural rules, we have called these not exponentials but *subexponentials* [19].

Proof theory does not provide canonical solutions for many things in computational logic: for example, the domain of first-order quantification is seldom addressed by proof theory as well as the exact nature of worlds within, say, Kripke models. These non-canonical aspects of logic provide, however, important opportunities for computer scientists to attach structures that they need to logic. Since the exponentials are not canonical, maybe there are possible exploitations of such non-canonical exponentials in computer science. Nigam and Miller have provided a partial answer to this question. In particular, they have shown that rich forms of multiset computation can be supported using subexponentials [19] [12]. In particular, it is possible to specify various multisets with different locations (identified with different exponentials) and to test them for emptiness. In this way, linear logic with an array of subexponential can then be used to declaratively and faithfully specify a wide range of deterministic and non-deterministic algorithms.

6.7. A framework for proof systems

Participants: Vivek Nigam, Dale Miller.

In earlier work by Pimentel and Miller [58], it was clear that linear logic could be used to encode provability in classical and intuitionistic logics using simple and elegant linear logic theories. Recently, Nigam and Miller have extended that work to show that by using polarity and focusing within linear logic, it is possible to account for a range of proof systems, such as, for example, sequent calculus, natural deduction, tableaux, free deduction, etc. The initial work in this area by Nigam, Pimentel, and Miller only captured *relative completeness* whereas the most recent our these papers [56] [12] are able to capture a much more refined notion of “adequate encoding”, namely, inference rules in one system are captured exactly as (focused) inference rules in the linear logical framework. In particular, Nigam and Miller argue that linear logic can be used as a meta-logic to specify a range of object-level proof systems. In particular, they showed that by providing different *polarizations* within a *focused proof system* for linear logic, one can account for natural deduction (normal and non-normal), sequent proofs (with and without cut), and tableaux proofs. Armed with just a few, simple variations to the linear logic encodings, more proof systems can be accommodated, including proof system using generalized elimination and generalized introduction rules. In general, most of these proof systems are developed for both classical and intuitionistic logics. By using simple results about linear logic, they could also give simple and modular proofs of the soundness and relative completeness of all the proof systems considered.

6.8. Modular Systems for Modal Logics

Participant: Lutz Straßburger.

There are modal logics like S4 or K, for which it is rather straightforward to provide a cut-free sequent system, and there are others, like S5 for which this is difficult or impossible. We (in a joint work with Kai Brünnler, Univ. Bern) used “nested sequents” [30] (a generalization of hypersequents [24]) to give a completely modular account to the whole modal cube below S5. That is to say, we have cut-free sequent systems for the basic normal modal logics formed by any combination of the axioms d, t, b, 4, 5, such that each axiom has a corresponding rule and each combination of these rules is complete for the corresponding frame conditions. This result are published in [16].

6.9. Generic cut-elimination for Substructural Logics

Participant: Lutz Straßburger.

We were able (in a joint work with Agata Ciabattoni, TU Wien, and Kazushige Terui, Kyoto University) to make further progress in the development of a systematic and algebraic proof theory for nonclassical logics. Continuing the work of [33] we defined a hierarchy on Hilbert axioms in the language of classical linear logic without exponentials, and gave a systematic procedure for transforming axioms up to the level P'_3 of the hierarchy into inference rules in multiple-conclusion (hyper)sequent calculi, which enjoy cut-elimination under a certain condition. This allows a systematic treatment of logics which could not be dealt with in previous approaches. Our method also works as a heuristic principle for finding appropriate rules for axioms located at levels higher than P'_3 . The work is published in [17].

6.10. A Kleene Theorem for Forest Languages

Participant: Lutz Straßburger.

Kleene's theorem on the coincidence of the rational and the recognizable languages in a free monoid [47] is a fundamental result of theoretical computer science. In [20] we present a generalization of Kleene's theorem to forest languages, which are a generalization of tree languages. However, our result is not a generalization of the result by Thatcher and Wright on tree languages [64]. We proposed an alternative approach to the standard notion of rational (or regular) expression for tree languages. The main difference is that in our new notion we have only one concatenation operation and only one star-operation, instead of many different ones. This is achieved by considering forests instead of trees over a ranked alphabet, or, algebraically speaking, by considering cartesian categories instead of term-algebras. The main result is that in the free cartesian category the rational languages and the recognizable languages coincide.

6.11. Computation with meta-variables

Participant: Stéphane Lengrand.

Meta-variables are central in proof search mechanisms to represent incomplete proofs and incomplete objects. They are used in almost all implementations of proof-related software, yet their meta-theory remains less explored than that of complete proofs and objects such as in the λ -calculus.

In 2009, Stéphane Lengrand and Jamie Murdoch Gabbay have published in [13] a first proposal for a computational model taking these features into account.

This proposal, extending the λ calculus with a particular kind of meta-variables originating from nominal logic, is more sparing than previous approaches like *Higher-Order Abstract Syntax*, which explicitly represents *all potential* dependencies between incomplete objects (this leads to computational inefficiencies as potential dependencies that are not effectively used still incur a computational cost).

Lengrand and Gabbay's proposal is only a first step, as it does not have a neat theory of normal forms (i.e. output values). A more complete version of such a λ -calculus, with incomplete objects and arbitrary binding dependencies but also with better normalization properties, has been in development since.

6.12. Encoding constrained transition systems

Participants: Kaustuv Chaudhuri, Joëlle Despeyroux.

Joëlle Despeyroux and Kaustuv Chaudhuri have given an encoding of the synchronous stochastic π -calculus in a hybrid extension of intuitionistic linear logic (called HyLL). Precisely, they have shown that focused partial sequent derivations in the encoding are in bijection with stochastic traces. The modal worlds are used to represent the rates of stochastic interactions, and the connectives of hybrid logic are used to represent the constraints in the stochastic transition rules. These results have been submitted to a journal [31] and an extended report is available from HAL [21].

One of the most successful applications of the stochastic π -calculus has been in representing signal transduction networks in cellular biology. An interesting application of this work would therefore be the direct representations of biological processes in HyLL, the original motivation for this line of investigation. Furthermore, other stochastic systems can, at least in principle, be similarly encoded in HyLL, giving us the linguistic ability to compare and combine systems represented using different stochastic formalisms.

6.13. Cut-elimination in classical logic

Participant: Stefan Hetzl.

A central research topic of the team are the connections between proof theory and computation. A proof-theoretic method which is often used for modeling computational systems is cut-elimination. The computational aspects of this procedure are rather well understood for intuitionistic and linear logic while the situation in classical logic is less satisfactory. [26] shows that in the general case the number of possible computational interpretations of a classical proof increases as strongly as its computational power and thus provides a lower bound on the set computations encoded by proof. [46] on the other hand considers proof in classical first-order logic and Peano arithmetic and derives an upper bound which is characterized by a regular tree grammar. This grammar can be used as alternative algorithm for computing cut-free proofs and is therefore a contribution to the reduction of syntax in proof theory.

7. Other Grants and Activities

7.1. Actions nationales

7.1.1. *INFER: ANR on the Theory and Application of Deep Inference*

Participants: Dale Miller, Lutz Straßburger.

The ANR-project Blanc titled “INFER: Theory and Application of Deep Inference” that is coordinated by Lutz Straßburger has been accepted in September 2006. Besides Parsifal, the teams associated with this effort are represented by François Lamarche (INRIA-Loria) and Michel Parigot (CNRS-PPS). Among the list of theoretical problems there is the fundamental need for a theory of correct identification of proofs, and its corollary, the development of a really general and flexible approach to proof nets. A closely related problem is the extension of the Curry-Howard isomorphism to these new representations. Among the list of more practical problems to be consider is the question of strategy and complexity in proof search, in particular for higher order systems. These questions are intimately related to how proofs themselves are formulated in these systems. Given their common grounding in rewriting theory, the proposal plans to deepen the relationship between deep inference and well established techniques like deduction modulo and unification for quantifiers. The proposal also plans to explore the formulation and use of more “exotic” logical systems, for example, non-commutative logics, that have interesting applications, such as in linguistics and quantum computing.

7.1.2. *PSI: ANR on Proof Search control in Interaction with domain-specific methods*

Participant: Stéphane Lengrand.

Stephane Lengrand is the scientific leader of the ANR-project Jeunes chercheurs entitled “Proof Search control in Interaction with domain-specific methods”, which was accepted in April 2009. Other founding members are among the INRIA project-team “TypiCal” : G. Faure and A. Mahboubi. Since the project started, Ph.D. student has joined the project’s research effort, and funding is available for a one-year post-doc and a three-year Ph.D., both starting in September 2010.

7.1.3. *CPP: ANR on Confidence, Proofs, and Probabilities*

Participants: Ivan Gazeau, Dale Miller.

The ANR Blanc titled “CPP: Confidence, Proofs, and Probabilities” has started 1 October 2009. This grant brings together the following institutions and individuals: LSV (Jean Goubault-Larrecq), CEA LIST (Eric Goubault, Olivier Bouissou, and Sylvie Putot), INRIA Saclay (Catuscia Palamidessi, Dale Miller, and Stephane Gaubert), Supelec L2S (Michel Kieffer and Eric Walter), and Supelec SSE (Gilles Fleury and Daniel Poulton). This project proposes to study the joint use of probabilistic and formal (deterministic) semantics and analysis methods, in a way to improve the applicability and precision of static analysis methods on numerical programs. The specific long-term focus is on control programs, e.g., PID (proportional-integral-derivative) controllers or possibly more sophisticated controllers, which are heavy users of floating-point arithmetic and present challenges of their own. To this end, we shall benefit from case studies and counsel from Hispano-Suiza and Dassault Aviation, who will participate in this project, but preferred to remain formally non-members, for administrative reasons.

7.1.4. *Panda: ANR on Parallelism and Distribution Analysis*

Participant: Dale Miller.

The ANR Blanc titled “Panda: Parallelism and Distribution Analysis” has started 1 October 2009. This project brings together researchers from INRIA Saclay (Comète and Parsifal), CEA LIST, MeASI as well labs in Paris (LIPN, PPS, LSV, LIP, LAMA), and on the Mediterranean (LIF, IML, Airbus). Scientifically, this proposal deals with the validation of concurrent and distributed programs, which is difficult because the number of its accessible states is too large to be enumerated, and even the number of control points, on which any abstract collecting semantics is based, explodes. This is due to the great number of distinct scheduling of actions in legal executions. This adds up to the important size of the codes, which, because they are less critical, are more often bigger. The objective of this project is to develop theories and tools for tackling this combinatorial explosion, in order to validate concurrent and distributed programs by static analysis, in an efficient manner. Our primary interest lies in multithreaded shared memory systems. But we want to consider a number of other paradigms of computations, encompassing most of the classical ones (message-passing for instance as in POSIX or VXWORKS) as well as more recent ones.

7.2. Actions internationales

7.2.1. *REDO: Redesigning logical syntax*

Participants: Nicolas Guenot, Dale Miller, Lutz Straßburger, François Wirion.

The REDO project is an INRIA funded ARC between INRIA Nancy–Grand Est, the University of Bath, and INRIA Saclay–Île-de-France. It started in January 2009 and lasts 2 years. Coordinator is Lutz Straßburger.

7.2.2. *Slimmer: an INRIA and NSF funded international team*

Participants: David Baelde, Andrew Gacek, Dale Miller.

Slimmer stands for *Sophisticated logic implementations for modeling and mechanical reasoning* is an “Equipes Associées” with seed money from INRIA. This project is initially designed to bring together the Parsifal personnel and Gopalan Nadathur’s Teyjus team at the University of Minnesota (USA). Separate NSF funding for this effort has also been awarded to the University of Minnesota. We are planning to expand the scope of this project to include other French and non-French sites, in particular, Alwen Tiu (Australian National University), Elaine Pimentel (Universidade Federal de Minas Gerais, Brazil) and Brigitte Pientka (McGill University, Canada).

7.2.3. *REUSSI: Research Experiences for US Students at INRIA*

Participants: David Baelde, Dale Miller, Zachary Snow.

This is an NSF funded project that places students from USA graduate programs in Computer Science within INRIA sites for internships ranging from a couple to several months in duration. During the last three summers, we have used these funds to support the visit of graduate students from the University of Minnesota to the Parsifal team.

8. Dissemination

8.1. Services to the Scientific Community

8.1.1. *Prices and distinctions*

Alexis Saurin’s PhD dissertation titled “Une étude logique du contrôle” won the “Prix de thèse ASTI 2009” as well as the “Prix de thèse de l’Ecole Polytechnique”.

8.1.2. *Organization of Conferences and Workshops*

Lutz Straßburger organized the first meeting of the REDO project in Palaiseau at LIX, May 26-29, 2009.

Lutz Straßburger co-organized (with Michel Parigot, PPS) the workshop "Structures and Deduction 2009" in Bordeaux, July 20–24, 2009. The workshop was part of the ESSLLI'09 summer school.

Lutz Straßburger co-organized (with Paola Bruscoli and Alessio Guglielmi, Bath and Nancy) the second meeting of the REDO project in Nancy at the Loria, November 16-18, 2009.

Stéphane Lengrand and Dale Miller organized the 2009 International Workshop on "Proof Search in Type Theories", affiliated to the CADE international conference which took place in Montréal, Canada, in August 2009.

8.1.3. Editorial activity

Dale Miller has the following editorial duties.

- *Theory and Practice of Logic Programming*. Member of Advisory Board since 1999. Cambridge University Press.
- *ACM Transactions on Computational Logic (ToCL)*. Editor-in-chief (since July 2009) and area editor for *Proof Theory* (since 1999). Published by ACM.
- *Journal of Functional and Logic Programming*. Permanent member of the Editorial Board since 1996. MIT Press.
- *Journal of Logic and Computation*. Associate editor since 1989. Oxford University Press.

8.1.4. Participation in program committees

Dale Miller was a program committee member for the following conferences.

- GaLoP IV: Games for Logic and Programming Languages, 28 - 29 March, York, UK.
- ICALP 09: International Colloquium on Automata, Languages and Programming, Rhodes, Greece, July.
- LSFA 2009: Fourth Logical and Semantic Frameworks, with Applications, part of RDP 2009, 28 June-3 July, Brasília, Brazil.
- LAM 2009: Logics for Agents and Mobility, August, Los Angeles. A workshop associated to LICS09.
- CSL 2009: 18th Annual Conference of the European Association for Computer Science Logic, 7-11 September, Coimbra, Portugal.
- Workshop on Games, Dialogue and Interaction, 28-29 Sept, Université Paris 8.

8.1.5. Invited talks at Conferences or Workshops

Dale Miller has been invited to speak at the following meetings.

- Colloquium on Games, Dialogue and Interaction, Université Paris 8, 28-29 September 2009.
- LAM 2009: Logics for Agents and Mobility. A workshop associated to LICS09. 9-10 August 2009.

8.2. Teaching

Dale Miller co-teaches the course "Logique Linéaire et paradigmes logiques du calcul" in the masters program MPRI ("Master Parisien de Recherche en Informatique") (2004-2009). Miller also taught a graduate level course on "Proof systems for linear, intuitionistic, and classical logic." Dipartimento di Informatica, Università Ca' Foscari di Venezia, 15-24 April 2009.

Stéphane Lengrand teaches the course "Logique formelle et Programmation Logique" at the École d'ingénieur ESIEA (2009). He teaches lab sessions in the main computer science curriculum of the Ecole Polytechnique.

8.3. Evaluation

Dale Miller has been on the PhD jury for the following three students:

- Denis Cousineau, Ecole Polytechnique, 1 Dec 2009 (examinateur).
- Xiaochu Qi, Computer Science Department, University of Minnesota, 9 September 2009 (external examiner).
- Andrew Gacek, Computer Science Department, University of Minnesota, 8 September 2009 (external examiner).

8.4. Internship supervision

Miller is currently supervising Anne-Laure Schneider (Poupon) (master student from ENS Paris) on the topic of dialog games for classical logic and Anders Starcke Henriksen (PhD student from Copenhagen) on the application of focusing proof systems.

9. Bibliography

Major publications by the team in recent years

- [1] J. DESPEYROUX, P. LELEU. *Recursion over Objects of Functional Type*, in "Special issue of MSCS on 'Modalities in Type Theory'", vol. 11, n^o 4, August 2001.
- [2] J. DESPEYROUX, F. PFENNING, C. SCHÜRMAN. *Primitive Recursion for Higher-Order Abstract Syntax*, in "Theoretical Computer Science (TCS)", vol. 266, n^o 1-2, September 2001, p. 1–57.
- [3] R. DYCKHOFF, S. LENGAND. *Call-by-Value λ -calculus and LJQ*, in "Journal of Logic and Computation", vol. 17, n^o 6, 2007, p. 1109–1134.
- [4] F. LAMARCHE, L. STRASSBURGER. *Naming Proofs in Classical Propositional Logic*, in "Typed Lambda Calculi and Applications, TLCA 2005", P. URZYCZYN (editor), LNCS, vol. 3461, Springer-Verlag, 2005, p. 246–261.
- [5] F. LAMARCHE, L. STRASSBURGER. *From Proof Nets to the Free *-Autonomous Category*, in "Logical Methods in Computer Science", vol. 2, n^o 4:3, 2006, p. 1–44, <http://arxiv.org/pdf/cs.LO/0605054>.
- [6] C. LIANG, D. MILLER. *Focusing and Polarization in Linear, Intuitionistic, and Classical Logics*, in "Theoretical Computer Science", vol. 410, n^o 46, 2009, p. 4747–4768 US .
- [7] R. MCDOWELL, D. MILLER. *Reasoning with Higher-Order Abstract Syntax in a Logical Framework*, in "ACM Trans. on Computational Logic", vol. 3, n^o 1, 2002, p. 80–136, <http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/mcdowell01.pdf>.
- [8] R. MCDOWELL, D. MILLER, C. PALAMIDESSI. *Encoding transition systems in sequent calculus*, in "Theoretical Computer Science", vol. 294, n^o 3, 2003, p. 411–437, <http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/tcs97.pdf>.
- [9] D. MILLER, A. TIU. *A proof theory for generic judgments*, in "ACM Trans. on Computational Logic", vol. 6, n^o 4, October 2005, p. 749–783, <http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/tocl-nabla.pdf>.

- [10] L. STRASSBURGER. *On the Axiomatisation of Boolean Categories with and without Medial*, in "Theory and Applications of Categories", vol. 18, n^o 18, 2007, p. 536–601, <http://arxiv.org/abs/cs.LO/0512086>.

Year Publications

Doctoral Dissertations and Habilitation Theses

- [11] O. DELANDE. *Symmetric Dialogue Games in the Proof Theory of Linear Logic*, Ecole Polytechnique, October 2009, <http://www.lix.polytechnique.fr/~delande/thesis.xhtml>, Ph. D. Thesis FR .
- [12] V. NIGAM. *Exploiting non-canonicity in the sequent calculus*, Ecole Polytechnique, September 2009, http://www.lix.polytechnique.fr/~nigam/thesis/Vivek_Nigam_phd.pdf, Ph. D. Thesis FR .

Articles in International Peer-Reviewed Journal

- [13] M. GABBAY, STÉPHANE. LENGRAND. *The lambda-context Calculus*, in "Information and Computation", vol. 207, n^o 12, 2009, p. 1369–1400 GB .
- [14] C. LIANG, D. MILLER. *Focusing and Polarization in Linear, Intuitionistic, and Classical Logics*, in "Theoretical Computer Science", vol. 410, n^o 46, 2009, p. 4747–4768, <http://dx.doi.org/10.1016/j.tcs.2009.07.041> US.

International Peer-Reviewed Conference/Proceedings

- [15] D. BAELDE. *On the proof theory of regular fixed points*, in "TABLEAUX 09: Automated Reasoning with Analytic Tableaux and Related Methods", M. GIESE, A. WALLER (editors), LNAI, n^o 5607, 2009, p. 93–107, <http://www.lix.polytechnique.fr/~dbaelde/productions/pool/baelde09tableaux.pdf> NO .
- [16] K. BRÜNNLER, L. STRASSBURGER. *Modular Sequent Systems for Modal Logic*, in "TABLEAUX 09: Automated Reasoning with Analytic Tableaux and Related Methods", M. GIESE, A. WALLER (editors), LNAI, n^o 5607, Springer, 2009, p. 152–166, <http://www.iam.unibe.ch/~kai/Papers/2009mssml.pdf> CH .
- [17] A. CIABATTONI, L. STRASSBURGER, K. TERUI. *Expanding the Realm of Systematic Proof Theory*, in "Computer Science Logic, CSL'09", E. GRÄDEL, R. KAHLE (editors), Lecture Notes in Computer Science, vol. 5771, Springer, 2009, p. 163–178, http://dx.doi.org/10.1007/978-3-642-04027-6_14.
- [18] C. LIANG, D. MILLER. *A Unified Sequent Calculus for Focused Proofs*, in "LICS: 24th Symp. on Logic in Computer Science", 2009, p. 355–364, <http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/liang09lics.pdf> US .
- [19] V. NIGAM, D. MILLER. *Algorithmic specifications in linear logic with subexponentials*, in "ACM SIGPLAN Conference on Principles and Practice of Declarative Programming (PPDP)", 2009, p. 129–140, <http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/ppdp09.pdf> PT .
- [20] L. STRASSBURGER. *A Kleene Theorem for Forest Languages*, in "Language and Automata Theory and Applications, LATA'09", A. H. DEDIU, A.-M. IONESCU, C. MARTÍN-VIDE (editors), LNCS, vol. 5457, Springer, 2009, p. 715–727, http://dx.doi.org/10.1007/978-3-642-00982-2_61.

Research Reports

- [21] K. CHAUDHURI, J. DESPEYROUX. *A Hybrid Linear Logic for Constrained Transition Systems with Applications to Molecular Biology*, INRIA, 2009, <http://hal.inria.fr/inria-00402942/en/>, Research ReportFR.

References in notes

- [22] S. ABRAMSKY. *Computational Interpretations of Linear Logic*, in "Theoretical Computer Science", vol. 111, 1993, p. 3–57.
- [23] J.-M. ANDREOLI. *Logic Programming with Focusing Proofs in Linear Logic*, in "Journal of Logic and Computation", vol. 2, n^o 3, 1992, p. 297–347.
- [24] A. AVRON. *The method of hypersequents in the proof theory of propositional non-classical logics*, in "Logic: from foundations to applications: European logic colloquium", Clarendon Press, 1996, p. 1–32.
- [25] B. E. AYDEMIR, A. BOHANNON, M. FAIRBAIRN, J. N. FOSTER, B. C. PIERCE, P. SEWELL, D. VYTINIO-TIS, G. WASHBURN, S. WEIRICH, S. ZDANCEWIC. *Mechanized Metatheory for the Masses: The PoplMark Challenge*, in "Theorem Proving in Higher Order Logics: 18th International Conference", LNCS, Springer-Verlag, 2005, p. 50–65.
- [26] M. BAAZ, S. HETZL. *On the non-confluence of cut-elimination*, 2009, manuscript AT .
- [27] D. BAELDE. *A linear approach to the proof-theory of least and greatest fixed points*, Ecole Polytechnique, December 2008, Ph. D. Thesis FR .
- [28] D. BAELDE. *On the Expressivity of Minimal Generic Quantification*, in "LFMTP 2008: International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice", A. ABEL, C. URBAN (editors), 2008, p. 16–31, <http://hal.inria.fr/inria-00284186/en/US>.
- [29] N. D. BELNAP, JR.. *Display Logic*, in "Journal of Philosophical Logic", vol. 11, 1982, p. 375–417.
- [30] K. BRÜNNLER. *Deep Sequent Systems for Modal Logic*, in "Advances in Modal Logic", G. GOVERNATORI, I. HODKINSON, Y. VENEMA (editors), vol. 6, College Publications, 2006, p. 107–119, <http://www.aiml.net/volumes/volume6/Bruennler.ps>.
- [31] K. CHAUDHURI, J. DESPEYROUX. *A Hybrid Linear Logic for Constrained Transition Systems*, June 2009, Submitted FR .
- [32] K. CHAUDHURI, D. MILLER, A. SAURIN. *Canonical Sequent Proofs via Multi-Focusing*, in "Fifth IFIP International Conference on Theoretical Computer Science", G. AUSIELLO, J. KARHUMÄKI, G. MAURI, L. ONG (editors), IFIP International Federation for Information Processing, vol. 273, Boston: Springer, September 2008, p. 383–396, <http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/tcs08trackb.pdf> IT .
- [33] A. CIABATTONI, N. GALATOS, K. TERUI. *From axioms to analytic rules in nonclassical logics*, in "23th Symp. on Logic in Computer Science", IEEE Computer Society Press, 2008, p. 229–240.
- [34] O. DELANDE, D. MILLER, A. SAURIN. *Proof and refutation in MALL as a game*, May 2009, <http://dx.doi.org/10.1016/j.apal.2009.07.017>, Accepted to the Annals of Pure and Applied LogicFR.

- [35] J. DESPEYROUX, P. LELEU. *A modal λ -calculus with iteration and case constructs*, in "proceedings of the annual Types for Proofs and Programs seminar", Springer-Verlag LNCS 1657, March 1998.
- [36] J. DESPEYROUX, F. PFENNING, C. SCHÜRMAN. *Primitive Recursion for Higher-Order Abstract Syntax*, in "proceedings of the TLCA 97 Int. Conference on Typed Lambda Calculi and Applications, Nancy, France, April 2–4", P. DE GROOTE, J. R. HINDLEY (editors), Springer-Verlag LNCS 1210, April 1997, p. 147–163.
- [37] A. GACEK. *A Framework for Specifying, Prototyping, and Reasoning about Computational Systems*, University of Minnesota, 2009, Ph. D. Thesis.
- [38] A. GACEK, D. MILLER, G. NADATHUR. *Combining generic judgments with recursive definitions*, in "23th Symp. on Logic in Computer Science", F. PFENNING (editor), IEEE Computer Society Press, 2008, p. 33–44, <http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/lics08a.pdf> US .
- [39] A. GACEK, D. MILLER, G. NADATHUR. *Reasoning in Abella about Structural Operational Semantics Specifications*, in "LFMTP 2008: International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice", A. ABEL, C. URBAN (editors), 2008, p. 75–89, <http://arxiv.org/pdf/0804.3914.pdf> US .
- [40] A. GACEK, D. MILLER, G. NADATHUR. *A two-level logic approach to reasoning about computations*, November 2009, <http://arxiv.org/pdf/0911.2993.pdf>, Submitted 16 November US .
- [41] A. GACEK, D. MILLER, G. NADATHUR. *Nominal abstraction*, August 2009, <http://arxiv.org/abs/0908.1390>, Extended version of LICS 2008 paper. Submitted US .
- [42] J.-Y. GIRARD. *Linear Logic*, in "Theoretical Computer Science", vol. 50, 1987, p. 1–102.
- [43] J.-Y. GIRARD. *Proof Theory and Logical Complexity, Volume I*, Studies in Proof Theory, vol. 1, Bibliopolis, edizioni di filosofia e scienze, 1987.
- [44] A. GUGLIELMI. *A System of Interaction and Structure*, in "ACM Trans. on Computational Logic", vol. 8, n° 1, 2007.
- [45] A. GUGLIELMI, L. STRASSBURGER. *Non-commutativity and MELL in the Calculus of Structures*, in "Computer Science Logic, CSL 2001", L. FRIBOURG (editor), LNCS, vol. 2142, Springer-Verlag, 2001, p. 54–68.
- [46] S. HETZL. *On the form of witness terms*, 2009, Submitted.
- [47] S. C. KLEENE. *Representation of events in nerve nets and finite automata*, in "Automata Studies", C. E. SHANNON, J. MCCARTHY (editors), Princeton, N.J., 1956, p. 3–40.
- [48] F. LAMARCHE, L. STRASSBURGER. *Constructing free Boolean categories*, in "Proceedings of the Twentieth Annual IEEE Symposium on Logic in Computer Science (LICS'05)", 2005, p. 209–218.
- [49] STÉPHANE. LENGREND, R. DYCKHOFF, J. MCKINNA. *A sequent calculus for Type Theory*, in "Proceedings of the 15th Annual Conference of the European Association for Computer Science Logic (CSL'06)", Z. ESİK (editor), Lecture Notes in Computer Science, vol. 4207, Springer-Verlag, September 2006, p. 441–455.

- [50] P. MARTIN-LÖF. *Constructive Mathematics and Computer Programming*, in "Sixth International Congress for Logic, Methodology, and Philosophy of Science, Amsterdam", North-Holland, 1982, p. 153–175.
- [51] R. MCDOWELL, D. MILLER. *Cut-elimination for a logic with definitions and induction*, in "Theoretical Computer Science", vol. 232, 2000, p. 91–119.
- [52] R. MCDOWELL, D. MILLER. *A Logic for Reasoning with Higher-Order Abstract Syntax*, in "Proceedings, Twelfth Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland", G. WINSKEL (editor), IEEE Computer Society Press, July 1997, p. 434–445.
- [53] D. MILLER. *Forum: A Multiple-Conclusion Specification Logic*, in "Theoretical Computer Science", vol. 165, n^o 1, September 1996, p. 201–232.
- [54] D. MILLER, G. NADATHUR, F. PFENNING, A. SCEDROV. *Uniform Proofs as a Foundation for Logic Programming*, in "Annals of Pure and Applied Logic", vol. 51, 1991, p. 125–157.
- [55] D. MILLER, A. TIU. *A Proof Theory for Generic Judgments: An extended abstract*, in "Proc. 18th IEEE Symposium on Logic in Computer Science (LICS 2003)", IEEE, June 2003, p. 118–127, <http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/lics03.pdf>.
- [56] V. NIGAM, D. MILLER. *A framework for proof systems*, March 2009, Extended version of IJCAR08 paper. Submitted FR .
- [57] F. PFENNING, C. SCHÜRMAN. *System Description: Twelf — A Meta-Logical Framework for Deductive Systems*, in "16th Conference on Automated Deduction, Trento", H. GANZINGER (editor), LNAI, n^o 1632, Springer, 1999, p. 202–206.
- [58] E. PIMENTEL, D. MILLER. *On the specification of sequent systems*, in "LPAR 2005: 12th International Conference on Logic for Programming, Artificial Intelligence and Reasoning", Lecture Notes in Artificial Intelligence, n^o 3835, 2005, p. 352–366, <http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/lpar05.pdf>.
- [59] A. M. PITTS. *Nominal Logic, A First Order Theory of Names and Binding*, in "Information and Computation", vol. 186, n^o 2, 2003, p. 165–193.
- [60] E. P. ROBINSON. *Proof Nets for Classical Logic*, in "Journal of Logic and Computation", vol. 13, 2003, p. 777–797.
- [61] A. SAURIN. *Une étude logique du contrôle (appliquée à la programmation fonctionnelle et logique)*, Ecole Polytechnique, September 2008, Ph. D. Thesis FR .
- [62] L. STRASSBURGER, F. LAMARCHE. *On Proof Nets for Multiplicative Linear Logic with Units*, in "Computer Science Logic, CSL 2004", J. MARCINKOWSKI, A. TARLECKI (editors), LNCS, vol. 3210, Springer-Verlag, 2004, p. 145–159.
- [63] L. STRASSBURGER. *What could a Boolean category be?*, in "Classical Logic and Computation 2006 (Satellite Workshop of ICALP'06)", S. VAN BAKEL (editor), 2006, <http://www.lix.polytechnique.fr/~lutz/papers/medial-kurz.pdf>.

-
- [64] J. W. THATCHER, J. B. WRIGHT. *Generalized Finite Automata Theory with an Application to a Decision Problem of Second-Order Logic*, in "Math. Systems Theory", vol. 2, 1968, p. 57–81.
- [65] A. TIU, D. MILLER. *Proof Search Specifications of Bisimulation and Modal Logics for the π -calculus*, February 2009, <http://arxiv.org/abs/0805.2785>, Accepted by ACM ToCL AU .
- [66] A. TIU. *A Logic for Reasoning about Generic Judgments*, in "Int. Workshop on Logical Frameworks and Meta-Languages: Theory and Practice (LFMTP'06)", A. MOMIGLIANO, B. PIENKA (editors), 2006.
- [67] A. TIU. *On the Role of Names in Reasoning about λ -tree Syntax Specifications*, in "LFMTP 2008: International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice", A. ABEL, C. URBAN (editors), 2008, p. 32–46.