IN PARTNERSHIP WITH:
CNRS

Ecole Polytechnique

Activity Report 2011

Project-Team PARSIFAL

Proof search and reasoning with logic specifications

IN COLLABORATION WITH: Laboratoire d’informatique de l’école polytechnique (LIX)

RESEARCH CENTER
Saclay - Île-de-France

THEME
Programs, Verification and Proofs

Table of contents

Project-Team PARSIFAL

Keywords: Proof Theory, Automated Theorem Proving, Programming Languages, Logics

1. Members

Research Scientists

Dale Miller [Team Leader, Senior Researcher, INRIA Saclay – Île-de-France]
Kaustuv Chaudhuri [Junior Researcher, INRIA Saclay – Île-de-France]
Joëlle Despeyroux [Junior Researcher, INRIA Sophia Antipolis – Méditerranée]
Stéphane Lengrand [Junior Researcher, CNRS]
Lutz Straßburger [Junior Researcher, INRIA Saclay – Île-de-France, HdR]

Technical Staff

Quentin Heath [ADT BATT, started 1 September 2011]

PhD Students

Mahfuza Farooque [ANR Jeunes Chercheurs PSI, started 1 October 2010]
Ivan Gazeau [ANR Blanc CPP, started 1 October 2009]
Nicolas Guenot [Allocataire Ministère de la Recherche, started 1 October 2008]
Anne-Laure Schneider [Allocation Internationale de thèse Gaspard Monge, started 1 September 2010]
Hernán Vanzetto [Contrat doctoral INRIA via INRIA-MSR, started 1 November 2010]

Post-Doctoral Fellows

Beniamino Accattoli [INRIA, 1 September 2011 - 31 August 2012]
Willem Heijltjes [INRIA, 1 October 2011 - 30 September 2012]
Clément Houtmann [INRIA, 1 September 2010 - 31 August 2011]

Administrative Assistant

Christelle Lievin

2. Overall Objectives

2.1. Main themes

The aim of the Parsifal team is to develop and exploit *proof theory* and *type theory* in the specification and verification of computational systems.

- *Expertise*: the team conducts basic research in proof theory and type theory. In particular, the team is developing results that help with automated deduction and with the manipulation and communication of formal proofs.
- *Design*: based on experience with computational systems and theoretical results, the team develops new logical principles, new proof systems, and new theorem proving environments.
- *Implementation*: the team builds prototype systems to help validate basic research results.
- *Examples*: the design and implementation efforts are guided by examples of specification and verification problems. These examples not only test the success of the tools but also drive investigations into new principles and new areas of proof theory and type theory.

The foundational work of the team focuses on *structural* and *analytic* proof theory, *i.e.*, the study of formal proofs as algebraic and combinatorial structures and the study of proof systems as deductive and computational formalisms. The main focus in recent years has been the study of the *sequent calculus* and of the *deep inference* formalisms.

An important research question is how to reason about computational specifications that are written in a *relational* style. To this end, the team has been developing new approaches to dealing with induction, co-induction, and generic quantification. A second important question is of *canonicity* in deductive systems, *i.e.*, when are two derivations "essentially the same"? This crucial question is important not only for proof search, because it gives an insight into the structure and an ability to manipulate the proof search space, but also for the communication of *proof objects* between different reasoning agents such as automated theorem provers and proof checkers.

Important application areas currently include:

- Meta-theoretic reasoning on functional programs, such as terms in the λ-calculus
- Reasoning about behaviors in systems with concurrency and communication, such as the π-calculus, game semantics, *etc.*
- Combining interactive and automated reasoning methods for induction and co-induction
- Verification of distributed, reactive, and real-time algorithms that are often specified using modal and temporal logics
- Representing proofs as documents that can be printed, communicated, and checked by a wide range of computational logic systems.

2.2. Highlights

- Josh Hodas and Dale Miller won the 2011 LICS Test of Time Award for their 1991 paper titled "Logic programming in a fragment of intuitionistic linear logic."
- Dale Miller's proposal titled "ProofCert: Broad Spectrum Proof Certificates" submitted to the ERC Advanced Investigator Grant in 2011 was accepted and will be funded for 2012-1016.

3. Scientific Foundations

3.1. General overview

There are two broad approaches for computational specifications. In the *computation as model* approach, computations are encoded as mathematical structures containing nodes, transitions, and state. Logic is used to *describe* these structures, that is, the 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.

The *computation as deduction* approach, in contrast, expresses computations logically, using formulas, terms, types, and proofs as computational elements. Unlike the model approach, general logical apparatus such as cut-elimination or automated deduction becomes directly applicable as tools for defining, analyzing, and animating computations. Indeed, we can identify two main aspects of logical specifications that have been very fruitful:

- *Proof normalization*, which treats the state of a computation as a proof term and computation as normalization of the proof terms. General reduction principles such as β-reduction or cut-elimination are merely particular forms of proof normalization. Functional programming is based on normalization [44], and normalization in different logics can justify the design of new and different functional programming languages [27].
- *Proof search*, which views the state of a computation as a a structured collection of formulas, known as a *sequent*, and proof search in a suitable sequent calculus as encoding the dynamics of the computation. Logic programming is based on proof search [48], and different proof search strategies can be used to justify the design of new and different logic programming languages [47].

While the distinction between these two aspects is somewhat informal, it helps to identify and classify different concerns that arise in computational semantics. For instance, confluence and termination of reductions are crucial considerations for normalization, while unification and strategies are important for search. A key challenge of computational logic is to find means of uniting or reorganizing these apparently disjoint concerns.

An important organizational principle is structural proof theory, that is, the study of proofs as syntactic, algebraic and combinatorial objects. Formal proofs often have equivalences in their syntactic representations, leading to an important research question about *canonicity* in proofs – when are two proofs "essentially the same?" The syntactic equivalences can be used to derive normal forms for proofs that illuminate not only the proofs of a given formula, but also its entire proof search space. The celebrated *focusing* theorem of Andreoli [28] identifies one such normal form for derivations in the sequent calculus that has many important consequences both for search and for computation. The combinatorial structure of proofs can be further explored with the use of *deep inference*; in particular, deep inference allows access to simple and manifestly correct cut-elimination procedures with precise complexity bounds.

Type theory is another important organizational principle, but most popular type systems are generally designed for either search or for normalization. To give some examples, the Coq system [56] that implements the Calculus of Inductive Constructions (CIC) is designed to facilitate the expression of computational features of proofs directly as executable functional programs, but general proof search techniques for Coq are rather primitive. In contrast, the Twelf system [52] that is based on the LF type theory (a subsystem of the CIC), is based on relational specifications in canonical form (*i.e.*, without redexes) for which there are sophisticated automated reasoning systems such as meta-theoretic analysis tools, logic programming engines, and inductive theorem provers. In recent years, there has been a push towards combining search and normalization in the same type-theoretic framework. The Beluga system [53], for example, is an extension of the LF type theory with a purely computational meta-framework where operations on inductively defined LF objects can be expressed as functional programs.

The Parsifal team investigates both the search and the normalization aspects of computational specifications using the concepts, results, and insights from proof theory and type theory.

3.2. Design of two level-logic systems

The team has spent a number of years in designing a strong new logic that can be used to reason (inductively and co-inductively) on syntactic expressions containing bindings. This work has been published is a series of papers by McDowell and Miller [46] [45], Tiu and Miller [50] [57], and Gacek, Miller, and Nadathur [12] [38]. Besides presenting formal properties of these logic, these papers also documented a number of examples where this logic demonstrated superior approaches to reasoning about a number of complex formal systems, ranging from programming languages to the λ-calculus and π-calculus.

The team has also been working on three different prototype theorem proving system that are all related to this stronger logic. These systems are the following.

- Abella, which is an interactive theorem prover for the full logic.
- Bedwyr, which is a model checker for the "finite" part of the logic.
- Tac, which is a sophisticate tactic for automatically completing simple proofs involving induction and unfolding.

We are now in the process of attempting to make all of these system communicate properly. Given that these systems have been authored by different team members at different times and for different reasons, they do not formally share the same notions of syntax and proof. We are now working to revisit all of these systems and revise them so that they all work on the *same* logic and so that they can share their proofs with each other.

Currently, Chaudhuri, Miller, and Accattoli are working with our technical staff member, Heath, to redesign and restructure these systems so that they can cooperate in building proofs.

3.3. Making the case for proof certificates

The team has been considering how it might be possible to define a universal format for proofs so that any existing theorem provers can have its proofs trusted by any other prover. This is a rather ambitious project and involves a great deal of work at the infrastructure level of computational logic. As a result, we have put significant energies into considering the high-level objectives and consequences of deploying such proof certificates.

Our current thinking on this point is roughly the following. Proofs, both formal and informal, are documents that are intended to circulate within societies of humans and machines distributed across time and space in order to provide trust. Such trust might lead a mathematician to accept a certain statement as true or it might help convince a consumer that a certain software system is secure. Using this general definition of proof, we have re-examined a range of perspectives about proofs and their roles within mathematics and computer science that often appears contradictory.

Given this view of proofs as both document and object, that need to be communicated and checked, we have attempted to define a particular approach to a *broad spectrum proof certificate* format that is intended as a universal language for communicating formal proofs among computational logic systems. We identify four desiderata for such proof certificates: they must be

1. checkable by simple proof checkers,
2. flexible enough that existing provers can conveniently produce such certificates from their internal evidence of proof,
3. directly related to proof formalisms used within the structural proof theory literature, and
4. permit certificates to elide some proof information with the expectation that a proof checker can reconstruct the missing information using bounded and structured proof search.

We consider various consequences of these desiderata, including how they can mix computation and deduction and what they mean for the establishment of marketplaces and libraries of proofs. More specifics can be found in Miller's papers [20] and [23].

3.4. Combining Classical and Intuitionistic Proof Systems

In order to develop an approach to proof certificates that is as comprehensive as possible, one needs to handle theorems and proofs in both classical logic and intuitionistic logic. Yet, building two separate libraries, one for each logic, can be inconvenient and error-prone. An ideal approach would be to design a single proof system in which both classical and intuitionistic proofs can exist together. Such a proof system should allow cut-elimination to take place and should have a sensible semantic framework.

Liang and Miller have recently been working on exactly that problem. In their paper [14], they showed how to describe a general setting for specifying proofs in intuitionistic and classical logic and to achieve one framework for describing initial-elimination and cut-elimination for such these two logics. That framework allowed for some mixing of classical and intuitionistic features in one logic. A more ambitious merging of these logics was provided in their work on "polarized intuitionistic logic" in which classical and intuitionistic connectives can be used within the same formulas [22].

3.5. Deep inference

Deep inference [40], [42] 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.

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.

3.6. Proof nets and atomic flows

Proof nets and atomic flows are abstract (graph-like) presentations of proofs such that all "trivial rule permutations" are quotiented away. Ideally the notion of proof net should be independent from any syntactic formalism, but 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 [39] for linear logic but also in Robinson’s proof nets [54] for classical logic. In this kind of proof nets every link in the net corresponds to a rule application in the sequent calculus.

Only recently, due to the rise of deep inference, new kinds of proof nets have been introduced that take the formula trees of the conclusions and add additional “flow-graph” information (see e.g., [6], [5] and [41]. On one side, this gives new insights in the essence of proofs and their normalization. But on the other side, all the known correctness criteria are no longer available.

This directly leads to the following research questions investigated by members of the parsifal team:

- 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 [6] this takes exponential time (in the size of the net).
- Studying the normalization of proofs in classical logic using atomic flows. Although there is no correctness criterion they allow to simplify the normalization procedure for proofs in deep inference, and additionally allow to get new insights in the complexity of the normalization.

4. Application Domains

4.1. Automated theorem proving

Automated theorem proving has traditionally focused on classical first-order logic, but non-classical logics are increasingly becoming important in the specification and analysis of software. Most type systems are based on (possibly second-order) propositional intuitionistic logic, for example, while resource-sensitive and concurrent systems are most naturally expressed in linear logic.

The members of the Parsifal team have a strong expertise in the design and implementation of performant automated reasoning systems for such non-classical logics. In particular, the Linprover suite of provers [34] continue to be the fastest automated theorem provers for propositional and first-order linear logic.

Any non-trivial specification, of course, will involve theorems that are simply too complicated to prove automatically. It is therefore important to design semi-automated systems that allow the user to give high level guidance, while at the same time not having to write every detail of the formal proofs. High level proof languages in fact serve a dual function – they are more readily comprehended by human readers, and they tend to be more robust with respect to maintenance and continued evolution of the systems. Members of the Parsifal team, in association with other INRIA teams and Microsoft Research, have been building a heterogeneous semi-automatic proof system for verifying distributed algorithms [36].

On a more foundational level, the team has been developing many new insights into the structure of proofs and the proof search spaces. Two directions, in particular, present tantalizing possibilities:

- The concept of *multi-focusing* [37] can be used to expose concurrency in computational behavior, which can in turn be exploited to prune areas of the proof search space that explore irrelevant interleavings of concurrent actions.
- The use of *bounded search*, where the bounds can be shown to be complete by meta-theoretic analysis, can be used to circumvent much of the non-determinism inherent in resource-sensitive logics such as linear logic. The lack of proofs of a certain bound can then be used to justify the presence or absence of properties of the encoded computations.

Much of the theoretical work on automated reasoning has been motivated by examples and implementations, and the Parsifal team intends to continue to devote significant effort in these directions.

4.2. Mechanized metatheory

There has been increasing interest in the use of formal methods to provide proofs of properties of programs and programming languages. Tony Hoare's Grand Challenge titled "Verified Software: Theories, Tools, Experiments" has as a goal the construction of "verifying compilers" for 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 [29] 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."

The Parsifal team has developed several tools and techniques for reasoning about the meta-theory of programming languages. One of the most important requirements for programming languages is the ability to reason about data structures with binding constructs up to α-equivalence. The use of higher-order syntax and nominal techniques for such data structures was pioneered by Miller, Nadathur and Tiu. The Abella system (see Section 3.2) implements a refinement of a number of these ideas and has been used to give full solutions to sections of the POPLMark challenge in addition to fully formal proofs of a number of other theorems in the meta-theory of the λ-calculus.

Now that the Abella system has been in circulation among colleagues during the past couple of years, there are many aspects of the methodology that now need to be addressed. During the summer of 2011, the team employed three interns Carnegie Mellon University and McGill University to work on different aspects of Abella. Particular focus was given to better ways to manipulate specification-logic contexts in the reasoning-logic and with finding ways to have Abella output a proper proof object (different from the scripts that are used to find a proof).

Our colleague Alwen Tiu from the Australian National University has also been building on our Bedwyr model checking tool so that we can build on top of it his SPEC system for doing model checking of spi-calculus expressions. We have adopted his enhancements to Bedwyr and are developing further improvements within the context of the BATT project (see Section 5.3).

4.3. Proof certificates

Members of the Parsifal team have shown how to specify a large variety of proof systems—including natural deduction, the sequent calculus, and various tableau and free deduction systems—uniformly using either focused linear logic [51], [49] or focused intuitionistic logic [43] as the meta-language. In the presence of induction and co-induction, arbitrary finite computations can be embedded into single synthetic steps [30]. Additional work [20] shows that this same framework can also capture resolution refutations as well as Pratt primality certificates.

An important application then of this work in designing synthetic inference systems based on classical and intuitionistic logic is that of designing a *broad spectrum proof certificate*. The definition of proof certificates can be remarkably flexible within the simple setting of focused proofs.

The most important implications of such a certificate format would be that most of the worlds theorem provers should be able to print out their proofs and communication them to other provers: these other provers could then check such certificates by expanding the synthetic connectives they contain down into a small and fixed set of “micro” inference rules.

5. Software

5.1. Profound

Participants: Kaustuv Chaudhuri [correspondant], Nicolas Guenot, Lutz Straßburger.

Profound is an interactive proof-development tool based on the focused calculus of structures [19]. It allows the user to build proofs using direct manipulation of the current proof state using the cursor keys and the mouse, instead of learning a formal textual proof interaction language. The tool checks all user actions dynamically with the aid of a theorem prover.

We plan to investigate adaptations of a tool such as Profound for proof development in other interactive proof development systems such as Abella or Coq. We also plan to use the high degree of proof compression that is enabled by the calculus of structures to create efficient proof certificates for exchange between different proof development systems.

The first public release of Profound is expected in December 2011. The development can be followed on INRIA GForge.

5.2. Abella

Participants: Andrew Gacek, Dale Miller.

The earliest versions of the Abella theorem prover was written while Gacek was a PhD student at the University of Minnesota. Two years ago, Gacek was a post doc in the Parsifal team and more features were added to this prover. During 2011, Chaudhuri and three interns (Andrew Cave from McGill and Salil Joshi and Chris Martens from CMU) developed some news designs and new prototypes of needed features for Abella. These features will provide this prover with better ways to manipulate specification-logic contexts in the reasoning-logic and a means for outputting proper proof object (different from the scripts that are used to find a proof).

For more information, see the Abella home page: http://abella.cs.umn.edu/.

5.3. Bedwyr

Participants: Kaustuv Chaudhuri, Quentin Heath, Dale Miller [correspondant].

During 2011, our close colleagues Alwen Tui (Australian National University) and David Baelde (INRIA team Proval) have made some improvements to the Bedwyr system. In the case of Tui, he made these changes in order to build SPEC, a model checker for the spi-calculus, on top of Bedwyr.

Starting in September, Quentin Heath has joined the team as a technical staff member. He is currently working on the Bedwyr code so that it can share files with the Abella system. These two provers work within a logic that is essentially the same: Heath is working to ensure that the concrete syntax and static semantics for the logical expressions on which they work is also the same. Thus, we expect to have our model checker (Bedwyr) and interactive theorem prover (Abella) share theories and proofs.

The work of Heath is being done in the context of the BATT ADJ project funded by INRIA. The boarder goals of the BATT project is to get four software systems (Bedwyr, Abella, Tac, and Teyjus) to inter-operate.

See also the web page http://slimmer.gforge.inria.fr/bedwyr/.

6. New Results

6.1. The Focused Calculus of Structures

Participants: Kaustuv Chaudhuri, Nicolas Guenot, Lutz Straßburger.

The *sequent calculus* is a proof system for logic that has many nice properties from a proof search perspective, the most famous being the subformula property that is essential to tame the search space. In recent years, the *focusing* property of sequent systems has become another useful property, both for shrinking the search space and to improve the representation of proofs. However, the sequent calculus does have some limitations. Primarily, not all logics have *analytic* (*i.e.*, cut-free) proof systems, which are the *sine qua non* of proof search. A less obvious but equally bothersome limitation is that cut-free sequent proofs tend to contain large repeated sub-proofs.

To remedy these deficiencies, one can use the *calculus of structures*, a proof system that allows inferences anywhere inside a formula. This system can represent many more logics than the sequent calculus and can produce better (*i.e.*, usually smaller) proofs because it can avoid sharing large subformulas. Nevertheless, because the rules of the calculus of structures have finer granularity than sequent rules, it has more non-determinism during search.

In this work, we show how to transplant the focusing result from the sequent calculus to the calculus of structures [19]. We thus improve the search capabilities of the calculus of structures, including the ability to go back and forth between focused sequent proofs and focused calculus of structures proofs, but we retain all the distinguishing features of the calculus of structures.

In particular, we preserve the ability to permute contractions below all other rules (first observed for the calculus of structures in [55], [31]). This permutation enables a two-stage normal form of proofs. The first stage contains only contractions, which increases the complexity of the formulas and is therefore a potential source of unbounded search; this phase needs to be recorded in order to reconstruct proofs by bounded search. The second stage that contains the remaining logical rules (except contraction) is strictly bounded and finite—hence decidable—and can be reconstructed if omitted from the proof object. Thus, we have the potential of obtaining very simple proof objects, recording only the first phase, for focused proofs; moreover, because of the bidirectional link, we can reconstruct focused sequent proofs from such proof objects.

Both the search and the representational aspects of focused calculus of structures proofs are being investigated in the Profound tool (see section 5.1).

6.2. Nested Sequents for Intuitionistic Modal Logics

Participant: Lutz Straßburger.

We show how the recent results for treating classical modal logics in the modal cube under S5 via nested sequents [32], [33] can be carried to intuitionistic modal logics. Thus, we present cut-free nested sequent systems for all intuitionistic modal logics in the modal cube up to IS5, and we show how this can be done in a modular way, i.e., to each of the axioms d, t, b, 4, and 5, we assign a set of inference rules, such that for each subset of d,t,b,4,5 the corresponding set of rules is sound and complete for the defined logic. This work has been presented in an invited talk at the IMLA workshop in Nancy [25].

6.3. The logic of nominal abstraction

Participant: Dale Miller.

In the paper [12], Gacek, Miller, and Nadathur have developed a strong logic that allows strong forms of induction and co-induction in the presence of the ∇ quantifier. This quantifier was introduced in earlier work by Miller and Tiu [50] where it was shown to provide declarative and flexible operational semantic specifications for a number of systems such as the λ-calculus and the π-calculus. The paper In the paper [12] introduces a generalization form of equality, called *nominal abstraction*, that permits natural specifications of predicates such as *freshness*. This paper provides the necessary meta theory (cut-elimination) for this new logic.

6.4. A framework for focusing and cut-elimination

Participant: Dale Miller.

Liang and Miller provide in [14] a general setting for specifying proofs in intuitionistic and classical logic. In this setting, it is possible to guarantee cut-elimination and initial elimination results simply by checking how certain simple side-conditions are used within a focused proof system. This setting treats both intuitionistic and classical logics as well allowing certain hybridizations of these two logics. This work helped lead the authors to finding a way to truly mix in one logic and one proof system both classical and intuitionistic logic, as described in [22].

6.5. Synthetic connectives and their proof system

Participant: Dale Miller.

In recent years, *focused proof systems* have being used to expand our understanding of how introduction rules and structural rule relate to each other. In these proof systems, inference rules and logical connectives are polarized as negative or positive in such a way that the invertible inference rules all belong to the negative polarity. Groups of negative connectives can then be grouped into one negative synthetic connective: similarly, positive connectives can be grouped into a positive synthetic connective. Such synthetic connectives admit cut-elimination. Remarkably, focused proof systems for classical and intuitionistic logics can be organized so that negative formulas are, in fact, treated linearly. That is, if weakening or contraction is applied to a formula, that formula is positive.

Focused proof systems can be used to design richly varying collections of synthetic connectives. These proof systems also provide for new means of describing parallelism within proofs and mixing computation and deduction. The ability to treat negative formulas linearly provides important information for the design of automated theorem provers. Synthetic connectives and their associated inference rules will also allow for the design of broad spectrum proof certificates that theorem provers will be able to print and simple proof checkers will be able to validate. Miller's conference paper [20] develops this approach to proof certificates.

6.6. Automated reasoning and SMT solving

Participants: Mahfuza Farooque, Stéphane Lengrand.

Automated reasoning uses a broad range of techniques whose soundness and completeness relate to the existence of proofs. The research programme of the ANR PSI project at Parsifal is to build a finer-grained connection by specifying automated reasoning techniques as the step-by-step construction of proofs, as we know it from proof theory and logic programming. The goal is to do this in a unifying framework, namely proof-search in a classical polarized sequent calculus. One of the advantages of this is to combine those techniques more easily. Another one is to envisage extending those techniques.

For instance, the algorithm at the heart of SMT-solving (SAT-modulo-Theory) is DPLL(T), whose theory does not treat existential variables (although SMT-provers often implement incomplete ad hoc techniques for them). We have first encoded DPLL(T) as the step-by-step construction of a proof tree in a classical polarized sequent calculus extended with calls to a decision procedure for T (to be published). This proof-theoretic view now allows us to envisage how to extend the algorithm with existential variables.

Another range of techniques that we are addressing is the handling of equality (superposition / paramodulation calculi).

6.7. Towards a Stochastic Linear Logic for Biological Computation

Participants: Kaustuv Chaudhuri, Joëlle Despeyroux.

In previous work [35], 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.

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.

This year, we have worked on specifying some simple examples of regulatory gene networks, together with basic properties of them—such as stability or oscillation—in HyLL. This is ongoing work in collaboration with Gilles Bernot's team at Nice-Sophia university.

7. Partnerships and Cooperations

7.1. National Initiatives

7.1.1. *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.

7.1.2. *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.

7.1.3. *PSI: ANR on Proof Search in Interaction with Domain-specific methods*

Participants: Stéphane Lengrand, Mahfuza Farooque.

The ANR Jeune Chercheuse / Jeune chercheur titled “PSI: Proof Search in Interaction with Domain-specific methods” has started 1 September 2009. This project investigates how proof-search can be performed in a framework where reasoning is subject to highly specific inference rules or axioms. This encompasses reasoning modulo a theory for which we may have a decision procedure (linear arithmetic, etc), or reasoning in a particular type theory (e.g. in a Pure Type system). The field of automated reasoning offers a variety of techniques (SAT-modulo-Theory, etc) which we like to see in terms of proof search. The project represent 192 000 euros of funding over four years, and is in collaboration with Assia Mahboubi at the TypiCal team.

7.2. International Initiatives

7.2.1. STRUCTURAL: ANR blanc International

Participants: Kaustuv Chaudhuri, Nicolas Guenot, Willem Heijltjes, Clément Houtmann, Dale Miller, Lutz Straßburger.

- Title: Structural and computational proof theory
- Duration: 01/01/2011 – 31/12/2013
- Partners:
  - University Paris VII, PPS (PI: Michel Parigot)
  - INRIA Saclay–IdF, EPI Parsifal (PI: Lutz Straßburger)
  - University of Innsbruck, Computational Logic Group (PI: Georg Moser)
  - Vienna University of Technology, Theory and Logic Group (PI: Matthias Baaz)
- Total funding by the ANR: 242 390,00 EUR (including 12 000 EUR pôle de compétivité: SYSTEMTIC Paris région)

This project is a consortium of four partners, two French and two Austrian, all being internationally recognized for their work on structural proof theory, but each coming from a different tradition. One of the objective of the project is build a bridge between these traditions and develop new proof-theoretic tools and techniques of structural proof theory having a strong potential of applications in computer science, in particular at the level of the models of computation and the extraction of programs and effective bounds from proofs.

On one side, there is the tradition coming from mathematics, which is mainly concerned with first-order logic, and studies, e.g., Herbrand’s theorem, Hilbert’s epsilon-calculus, and Goedel’s Dialectica interpretation. On the other side, there is the tradition coming from computer science, which is mainly concerned with propositional systems, and studies, e.g., Curry-Howard isomorphism, algebraic semantics, linear logic, proof nets, and deep inference. A common ground of both traditions is the paramount role played by analytic proofs and the notion of cut elimination. We will study the inter-connections of these different traditions, in particular we focus on different aspects and developments in deep inference, the Curry-Howard correspondence, term-rewriting, and Hilbert’s epsilon calculus. As a byproduct this project will yield a mutual exchange between the two communities starting from this common ground, and investigate, for example, the relationship between Herbrand expansions and the computational interpretations of proofs, or the impact of the epsilon calculus on proof complexity.

Besides the old, but not fully exploited, tools of proof theory, like the epsilon-calculus or Dialectica interpretation, the main tool for our research will be deep inference. Deep inference means that inference rules are allowed to modify formulas deep inside an arbitrary context. This change in the application of inference rules has drastic effects on the most basic proof theoretical properties of the systems, like cut elimination. Thus, much of the early research on deep inference went into reestablishing these fundamental results of logical systems. Now, deep inference is a mature paradigm, and enough theoretical tools are available to think to applications. Deep inference provides new properties, not available in shallow deduction systems, namely full symmetry and atomicity, which open new possibilities at the computing level that we intend to investigate in this project. We intend to investigate the precise relation between deep inference and term rewriting, and hope to develop a general theory of analytic calculi in deep inference. In this way, this project is a natural continuation of the ANR project INFER which ended in May 2010.

7.2.2. Eternal: INRIA ARC

Participants: Kaustuv Chaudhuri, Dale Miller, Lutz Straßburger.

Title: Interactive Resource Analysis

webpage: http://eternal.cs.unibo.it/

INRIA principal investigator: Dale Miller

INRIA Partner:

Institution: INRIA

Team: FOCUS

Researcher: Ugo Dal Lago

INRIA Partner:

Institution: INRIA

Team: pi.r2

Researcher: Pierre-Louis Curien

Duration: 2011 - 2013

This project aims at putting together ideas from Implicit Computational Complexity and Interactive Theorem Proving, in order to develop new methodologies for handling quantitative properties related to program resource consumption, like execution time and space. The task of verifying and certifying quantitative properties is undecidable as soon as the considered programming language gets close to a general purpose language. So, full-automatic techniques in general cannot help in classifying programs in a precise way with respect to the amount of resources used and moreover in several cases the programmer will not gain any relevant information on his programs. In particular, this is the case for all the techniques based on the study of structural constraints on the shape of programs, like many of those actually proposed in the field of implicit computational complexity. To overcome these limitations, we aim at combining the ideas developed in the linear logic approach to implicit computational complexity with the ones of interactive theorem proving, getting rid of the intrinsic limitations of the automatic techniques. In the obtained framework, undecidability will be handled through the system's user, who is asked not only to write the code, but also to drive the semi-automatic system in finding a proof for the quantitative properties of interest. In order to reduce the user effort and allow him to focus only on the critical points of the analysis, our framework will integrate implicit computational complexity techniques as automatic decision procedures for particular scenarios. Moreover, in order to be widely applicable, the modularity of the framework will permit to deal with programs written in different languages and to consider different computational resources. The kind of study proposed by this project has been almost neglected so far. Here, we aim at providing such a framework for both theoretic investigations and for testing in practice the effectiveness of the approach.

7.2.3. INRIA Associate Teams

7.2.3.1. RAPT

Participants: Beniamino Accattoli, Kaustuv Chaudhuri, Quentin Heath, Clément Houtmann, Dale Miller.

Title: Computational logic systems

INRIA principal investigator: Kaustuv Chaudhuri

International Partner:

Institution: McGill University (Canada)

Laboratory: School of Computer Science

Researcher: Brigitte Pientka

International Partner:

Institution: Carnegie Mellon University (United States)

Laboratory: Department of Computer Science

Researcher: Frank Pfenning

Duration: 2011 - 2013

See also: http://www.lix.polytechnique.fr/~kaustuv/rapt/

Many aspects of computation systems, ranging from operational semantics, interaction, and various forms of static analysis, are commonly specified using inference rules, which themselves are formalized as theories in a logical framework. While such a use of logic can yield sophisticated, compact, and elegant specifications, formal reasoning about these logic specifications presents a number of difficulties. The RAPT project will address the problem of reasoning about logic specifications by bringing together three different research teams, combining their backgrounds in type theory, proof theory, and the building of computational logic systems. We plan to develop new methods for specifying computation that allow for a range of specification logics (eg, intuitionistic, linear, ordered) as well as new means to reason inductively and co-inductively with such specifications. New implementations of reasoning systems are planned that use interactive techniques for deep meta-theoretic reasoning and fully automated procedures for a range of useful theorems.

7.2.4. Visits of International Scientists

7.2.4.1. Invited Researchers

- Alberto Momigliano, Associate Professor, University of Milan
  24 - 28 January and 30 - 31 August.
- Vivek Nigam, Research Scientist, LMU, Munich, Germany.
  26 April - 6 May.
- Chuck Liang, Professor, Hofstra University, NY, USA.
  2 June - 1 July
- Gopalan Nadathur, Professor, University of Minnesota, MN, USA.
  6 -10 June and 3 - 28 October.
- Elaine Pimentel, Associate Professor, Universidade Federal de Minas Gerais.
  13 - 24 June.
- Brigitte Pientka, Associate Professor, McGill University, Montreal, Canada.
  16 - 20 May.
- Alwen Tiu, Research Scientist, Australian National University.
  22 - 26 August.
- Anupam Das, PhD Student, University of Bath, UK.
  21 - 25 November 2011

7.2.4.2. Internships

- Andrew Cave, PhD student at McGill Univ., Montreal, Canada.
  Internship during May – July 2011
- Salil Joshi, PhD student at Carnegie Mellon Univ., USA.
  Internship during June – August 2011

- Chris Martens, PhD student at Carnegie Mellon Univ., USA.
  Internship during June – August 2011

7.2.5. Participation In International Programs

The team has travel funds within the following international programs.

- PHC Germaine de Staël 2011: funding travel between Bern, Switzerland and INRIA.
- 63.123 - 63ème CPCFQ: Commission permanente de coopération franco-québécoise: funding exchanges between the McGill and INRIA.
- INRIA-FAPEMIG: funding between INRIA and the Brazilian funding agency FAPEMIG located in the state of Minas Gerais.

8. Dissemination

8.1. Scientific Co-ordination

- During 2011, Dale Miller has served on the programme committees for LAM 2011 (Fourth International Workshop on Logics, Agents, and Mobility), 10 September, Aachen, Germany; MLPA-11 (Modules and Libraries for Proof Assistants), 26 August, Nijmegen; LICS 2011 (Logic in Computer Science), 21-24 June, Toronto; and Tableaux 2011 (20th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods), July 4-8, Bern, Switzerland.
- Stéphane Lengrand organized the 2011 workshop on “Proof Search in Type Theories and Axiomatic Theories”, affiliated to CADE’23, on 1st August, Wroclaw, Poland, and the Computational Logic workshop in honour of Roy Dyckhoff on 18th and 19th November, St Andrews, Scotland.
- Lutz Straßburger has served on the programme committee for the ESSLLI’11 student session, and has reviewed papers for the following journals: TCS, LMCS, RSL, and the following conferences: LICS’11, TLCA’11, TABLEAUX’11, APLAS’11, STACS’11.
- Stéphane Lengrand and Lutz Straßburger organized the joint meeting of the *GdT Geocal* and the *GdT LAC* of the *GDR Informatique Mathématique* from November 25 to November 26, 2011 at Ecole Polytechnique.
- Kaustuv Chaudhuri was the local chair of the first STRUCTURAL workshop held at LIX, École Polytechnique, in March 2011. He has reviewed papers for: JAR and ACM ToCL (journals) and CADE 2011, LICS 2011, CSL 2011, and LPAR 2012.

8.2. Evaluation

- Lutz Straßburger served as rapporteur on the PhD-jury for Novak Novakovic, Institut National Polytechnique de Lorraine (INPL), Nancy, November 8, 2011
- Dale Miller served as a rapporteur on the PhD jury for Anders Starcke-Henriksen, University of Copenhagen, 22 December, 2011.
- Miller was member of the PhD juries for François Garillot (Ecole Polytechnique, December 2011) and Daniel Weller (Technische Universität Wien, 12 January, 2011)

8.3. Teaching

Master : Dale Miller taught 12 hours at MPRI (Master Parisien de Recherche en Informatique) in the Course 2-1: Logique linéaire et paradigmes logiques du calcul.

Master : Dale Miller taught 20 hours in a graduate course at the University of Pisa for two weeks, 12 - 23 September.

Master : Dale Miller taught 6 hours in a graduate course at the ISCL: International School on Computational Logic, Bertinoro, 10-15 April 2011.

PhD & HdR :

HdR : Lutz Straßburger, “Towards a Theory of Proofs of Classical Logic” [11], Université Paris VII, January 7, 2011

PhD in progress :

1. Ivan Gazeau, since October 2009, co-supervisor: Dale Miller
2. Nicolas Guenot, “Nested Deduction as Foundation for Computation”, since September 2008, supervisor: Lutz Straßburger
3. Mahfuza Farooque, since October 2010, co-supervisor: Stéphane Lengrand
4. Anne-Laure Schneider, since September 2010, supervisor: Dale Miller
5. Hernán Vanzetto, since March 2011, co-supervisor: Kaustuv Chaudhuri

8.4. Popularization

- Lutz Straßburger gave a talk on “C’est quoi, une preuve ?” within the “Unité ou café” series at INRIA Saclay–IdF, March 11, 2011

9. Bibliography

Major publications by the team in recent years

[1] K. CHAUDHURI, F. PFENNING, G. PRICE. *A Logical Characterization of Forward and Backward Chaining in the Inverse Method*, in "J. of Automated Reasoning", March 2008, vol. 40, n^{o} 2-3, p. 133–177 [*DOI :* s10817-007-9091-0], http://www.springerlink.com/content/b335r17728862pm2/fulltext.pdf.

[2] O. DELANDE, D. MILLER, A. SAURIN. *Proof and refutation in MALL as a game*, in "Annals of Pure and Applied Logic", February 2010, vol. 161, n^{o} 5, p. 654–672, http://dx.doi.org/10.1016/j.apal.2009.07.017.

[3] R. DYCKHOFF, S. LENGRAND. *Call-by-Value λ-calculus and LJQ*, in "Journal of Logic and Computation", 2007, vol. 17, n^{o} 6, p. 1109–1134.

[4] A. GACEK, D. MILLER, G. NADATHUR. *Nominal abstraction*, in "Information and Computation", 2011, vol. 209, n^{o} 1, p. 48–73, http://arxiv.org/abs/0908.1390.

[5] A. GUGLIELMI, T. GUNDERSEN, L. STRASSBURGER. *Breaking Paths in Atomic Flows for Classical Logic*, in "Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS 2010)", Edinburgh, United Kingdom, July 2010, p. 284–293 [*DOI :* 10.1109/LICS.2010.12], http://www.lix.polytechnique.fr/~lutz/papers/AFII.pdf.

[6] F. LAMARCHE, L. STRASSBURGER. *Naming Proofs in Classical Propositional Logic*, in "Typed Lambda Calculi and Applications, TLCA 2005", P. URZYCZYN (editor), LNCS, Springer-Verlag, 2005, vol. 3461, p. 246–261.

[7] S. LENGRAND, R. DYCKHOFF, J. MCKINNA. *A focused sequent calculus framework for proof search in Pure Type Systems*, in "Logical Methods in Computer Science", 2010, To appear.

[8] C. LIANG, D. MILLER. *Focusing and Polarization in Linear, Intuitionistic, and Classical Logics*, in "Theoretical Computer Science", 2009, vol. 410, n^{o} 46, p. 4747–4768.

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

[10] A. TIU, D. MILLER. *Proof Search Specifications of Bisimulation and Modal Logics for the π-calculus*, in "ACM Trans. on Computational Logic", 2010, vol. 11, n^{o} 2, http://arxiv.org/abs/0805.2785.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[11] L. STRASSBURGER. *Towards a Theory of Proofs of Classical Logic*, Université Paris VII, 2011.

Articles in International Peer-Reviewed Journal

[12] A. GACEK, D. MILLER, G. NADATHUR. *Nominal abstraction*, in "Information and Computation", 2011, vol. 209, n^{o} 1, p. 48–73, http://arxiv.org/abs/0908.1390.

[13] A. GUGLIELMI, L. STRASSBURGER. *A system of interaction and structure V: the exponentials and splitting*, in "Mathematical Structures in Computer Science", 2011, vol. 21, n^{o} 3, p. 563–584.

[14] C. LIANG, D. MILLER. *A Focused Approach to Combining Logics*, in "Annals of Pure and Applied Logic", 2011, vol. 162, n^{o} 9, p. 679–697, http://dx.doi.org/10.1016/j.apal.2011.01.012.

[15] L. STRASSBURGER, A. GUGLIELMI. *A system of interaction and structure IV: The exponentials and decomposition*, in "ACM Trans. Comput. Log.", 2011, vol. 12, n^{o} 4, 23.

[16] L. STRASSBURGER. *From Deep Inference to Proof Nets via Cut Elimination*, in "J. Log. Comput.", 2011, vol. 21, n^{o} 4, p. 589–624.

International Conferences with Proceedings

[17] A. BERNADET, STÉPHANE. LENGRAND. *Complexity of strongly normalising λ-terms via non-idempotent intersection types*, in "Proceedings of the 14th international conference on Foundations of Software Science and Computation Structures (FOSSACS'11)", M. HOFMANN (editor), LNCS, Springer-Verlag, March 2011, vol. 6604, p. 88–107.

[18] A. BERNADET, STÉPHANE. LENGRAND. *Filter models: non-idempotent intersection types, orthogonality and polymorphism*, in "Proceedings of the 20th Annual conference of the European Association for Computer Science Logic (CSL'11)", M. BEZEM (editor), Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, September 2011.

[19] K. CHAUDHURI, N. GUENOT, L. STRASSBURGER. *The Focused Calculus of Structures*, in "Computer Science Logic: 20th Annual Conference of the EACSL", Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, September 2011, p. 159–173 [*DOI :* 10.4230/LIPICS.CSL.2011.159], http://drops.dagstuhl.de/opus/volltexte/2011/3229/pdf/16.pdf.

[20] D. MILLER. *A proposal for broad spectrum proof certificates*, in "CPP: First International Conference on Certified Programs and Proofs", J.-P. JOUANNAUD, Z. SHAO (editors), 2011.

Other Publications

[21] A. BRUNEL, O. HERMANT, C. HOUTMANN. *Orthogonality and Boolean Algebras for Deduction Modulo*, http://hal.inria.fr/inria-00563331/en.

[22] C. LIANG, D. MILLER. *Kripke Semantics and Proof Systems for Combining Intuitionistic Logic and Classical Logic*, September 2011, Submitted.

[23] D. MILLER. *Communicating and trusting proofs: The case for broad spectrum proof certificates*, June 2011, Available from author's website.

[24] L. MÉHATS, L. STRASSBURGER. *Non-crossing tree realizations of ordered degree sequences*, http://hal.inria.fr/hal-00649591.

[25] L. STRASSBURGER. *Nested sequent systems for intuitionistic modal logics*, 2011, Invited talk at the workshop "Intuitionistic Modal Logics and Applications" (IMLA'11), part of 14th Congress of Logic, Methodology and Philosophy of Science, Nancy, July 19–26, 2011.

[26] L. STRASSBURGER. *Towards a Theory of Proofs of Classical Logic*, 2011, Invited talk at the workshop "Proof Systems at the Test of Computer Science: Foundational and Applicational Encounters", part of 14th Congress of Logic, Methodology and Philosophy of Science, Nancy, July 19–26, 2011.

References in notes

[27] S. ABRAMSKY. *Computational Interpretations of Linear Logic*, in "Theoretical Computer Science", 1993, vol. 111, p. 3–57.

[28] J.-M. ANDREOLI. *Logic Programming with Focusing Proofs in Linear Logic*, in "Journal of Logic and Computation", 1992, vol. 2, n^o 3, p. 297–347.

[29] B. E. AYDEMIR, A. BOHANNON, M. FAIRBAIRN, J. N. FOSTER, B. C. PIERCE, P. SEWELL, D. VYTINIOTIS, 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.

[30] D. BAELDE, D. MILLER, Z. SNOW. *Focused Inductive Theorem Proving*, in "Fifth International Joint Conference on Automated Reasoning (IJCAR 2010)", J. GIESL, R. HÄHNLE (editors), LNCS, 2010, n^o 6173, p. 278–292 [*DOI :* 10.1007/978-3-642-14203-1], http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/ijcar10.pdf.

[31] K. BRÜNNLER. *Deep Inference and Symmetry for Classical Proofs*, Technische Universität Dresden, 2003.

[32] K. BRÜNNLER. *Deep Sequent Systems for Modal Logic*, in "Advances in Modal Logic", G. GOVERNATORI, I. HODKINSON, Y. VENEMA (editors), College Publications, 2006, vol. 6, p. 107–119, http://www.aiml.net/volumes/volume6/Bruennler.ps.

[33] K. BRÜNNLER, L. STRASSBURGER. *Modular Sequent Systems for Modal Logic*, in "Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX'09", M. GIESE, A. WAALER (editors), LNCS, Springer, 2009, vol. 5607, p. 152–166.

[34] K. CHAUDHURI. *The Focused Inverse Method for Linear Logic*, Carnegie Mellon University, December 2006, Technical report CMU-CS-06-162, http://reports-archive.adm.cs.cmu.edu/anon/2006/CMU-CS-06-162.pdf.

[35] K. CHAUDHURI, J. DESPEYROUX. *A Hybrid Linear Logic for Constrained Transition Systems with Applications to Molecular Biology*, INRIA, 2010, http://hal.inria.fr/inria-00402942/.

[36] K. CHAUDHURI, D. DOLIGEZ, L. LAMPORT, S. MERZ. *Verifying Safety Properties With the TLA$^+$ Proof System*, in "Fifth International Joint Conference on Automated Reasoning (IJCAR 2010)", Edinburgh, United Kingdom, J. GIESL, R. HÄHNLE (editors), LNAI, Springer, July 2010, vol. 6173, p. 142–148 [*DOI :* 10.1007/978-3-642-14203-1_12], http://hal.inria.fr/inria-00534821.

[37] 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, Boston: Springer, September 2008, vol. 273, p. 383–396, http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/tcs08trackb.pdf.

[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.

[39] J.-Y. GIRARD. *Linear Logic*, in "Theoretical Computer Science", 1987, vol. 50, p. 1–102.

[40] A. GUGLIELMI. *A System of Interaction and Structure*, in "ACM Trans. on Computational Logic", 2007, vol. 8, n^o 1.

[41] A. GUGLIELMI, T. GUNDERSEN. *Normalisation Control in Deep Inference Via Atomic Flows*, in "Logical Methods in Computer Science", 2008, vol. 4, n^o 1:9, p. 1–36, http://arxiv.org/abs/0709.1205.

[42] A. GUGLIELMI, L. STRASSBURGER. *Non-commutativity and MELL in the Calculus of Structures*, in "Computer Science Logic, CSL 2001", L. FRIBOURG (editor), LNCS, Springer-Verlag, 2001, vol. 2142, p. 54–68.

[43] A. S. HENRIKSEN. *Using LJF as a Framework for Proof Systems*, University of Copenhagen, 2009, http://hal.inria.fr/inria-00442159/en/.

[44] 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.

[45] R. MCDOWELL, D. MILLER. *Reasoning with Higher-Order Abstract Syntax in a Logical Framework*, in "ACM Trans. on Computational Logic", 2002, vol. 3, n^o 1, p. 80–136, http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/mcdowell01.pdf.

[46] 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.

[47] D. Miller. *Forum: A Multiple-Conclusion Specification Logic*, in "Theoretical Computer Science", September 1996, vol. 165, n^o 1, p. 201–232.

[48] D. Miller, G. Nadathur, F. Pfenning, A. Scedrov. *Uniform Proofs as a Foundation for Logic Programming*, in "Annals of Pure and Applied Logic", 1991, vol. 51, p. 125–157.

[49] D. Miller, E. Pimentel. *Using linear logic to reason about sequent systems*, in "International Conference on Automated Reasoning with Analytic Tableaux and Related Methods", U. Egly, C. Fermüller (editors), LNCS, Springer, 2002, vol. 2381, p. 2–23, http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/tableaux02.pdf.

[50] 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.

[51] V. Nigam, D. Miller. *A framework for proof systems*, March 2009, Extended version of IJCAR08 paper. Submitted.

[52] F. Pfenning, C. Schürmann. *System Description: Twelf — A Meta-Logical Framework for Deductive Systems*, in "16th Conference on Automated Deduction", Trento, H. Ganzinger (editor), LNAI, Springer, 1999, n^o 1632, p. 202–206.

[53] B. Pientka, J. Dunfield. *Beluga: A Framework for Programming and Reasoning with Deductive Systems (System Description)*, in "Fifth International Joint Conference on Automated Reasoning", J. Giesl, R. Hähnle (editors), LNCS, 2010, n^o 6173, p. 15–21.

[54] E. P. Robinson. *Proof Nets for Classical Logic*, in "Journal of Logic and Computation", 2003, vol. 13, p. 777–797.

[55] L. Strassburger. *Linear Logic and Noncommutativity in the Calculus of Structures*, Technische Universität Dresden, 2003.

[56] The Coq Development Team. *The Coq Proof Assistant Version 8.3 Reference Manual*, INRIA, October 2010.

[57] A. Tiu, D. Miller. *Proof Search Specifications of Bisimulation and Modal Logics for the π-calculus*, in "ACM Trans. on Computational Logic", 2010, vol. 11, n^o 2, http://arxiv.org/abs/0805.2785.