IN PARTNERSHIP WITH:
CNRS

Ecole Polytechnique

Activity Report 2013

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
Proofs and Verification

Table of contents

Project-Team PARSIFAL

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

Creation of the Project-Team: 2007 July 01.

1. Members

Research Scientists

Dale Miller [Team leader, Inria, Senior Researcher, HdR]
Kaustuv Chaudhuri [Inria, Researcher]
François Lamarche [Inria, Senior Researcher, HdR]
Lutz Straßburger [Inria, Researcher, HdR]
Stéphane Graham-Lengrand [CNRS, Researcher]

PhD Students

Hichem Chihani [Inria, FP7 ERC PROOFCERT grant, from Oct 2012]
Mahfuza Farooque [CNRS, until Dec 2013]
Ivan Gazeau [ATER Université Paris XI, until Oct 2013]
Quentin Heath [Ecole Polytechnique, from Oct 2013]

Post-Doctoral Fellows

Ryuta Arisaka [Inria, ANR STRUCTURAL INTB grant, from May 2013]
Matteo Cimini [Inria, ANR STRUCTURAL INTB grant]
Anupam Das [Inria, from Nov 2013]
Stefan Hetzl [Inria, ANR STRUCTURAL INTB and PARSIFAL, until Jan 2013]
Danko Ilik [Inria, FP7 ERC PROOFCERT grant, from Dec 2013]
Novak Novakovic [Inria, ANR STRUCTURAL INTB grant, until Nov 2013]
Fabien Renaud [Inria, FP7 ERC PROOFCERT grant]
Mikheil Rukhaia [Inria, ANR STRUCTURAL INTB grant, Apr–Dec 2013]

Visiting Scientists

Chuck Liang [Hofstra University, Jun 2013]
Stefan Mehner [U. Bonn, Dec 2013]
Gopalan Nadathur [Univ. of Minnesota, May 2013]
Elaine Pimentel [UFRN, Brazil, Jun–Jul 2013]
Thanos Tsouanas [ENS Lyon, Jan 2013]

Administrative Assistant

Christelle Liévin [Inria]

Others

Olivier Savary-Bélanger [McGill University, internship, May–August 2013]
Stéphane Bersier [Ecole Polytechnique, internship, Apr–Aug 2013]
Anastasia Gkolfi [Inria, internship, Apr–Aug 2013]
Sonia Marin [Inria, internship, Jun–Aug 2013]
Jean Pichon [Inria, internship, Mar–Aug 2013]

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 of the Year

- The team organized the LIX Colloquium 2013: The Theory and Application of Formal Proofs, November 5–7. Webpage: http://www.lix.polytechnique.fr/colloquium2013/

3. Research Program

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 [48], and normalization in different logics can justify the design of new and different functional programming languages [35].
- *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 [53], and different proof search strategies can be used to justify the design of new and different logic programming languages [52].

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 [36] 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 [59] 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 [55] 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 [56], 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 [50] [49], Tiu and Miller [54] [61], and Gacek, Miller, and Nadathur [2] [41]. 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.

During 2013, Chaudhuri and Miller worked 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 is developing a framework for describing the semantics of proof evidence so that any existing theorem prover can have its proofs trusted by any other prover. This is an 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 [8] and [51].

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 [7], 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 [16].

3.5. Deep inference

Deep inference [43], [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.

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 [42] for linear logic but also in Robinson’s proof nets [57] 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., [4], [3] and [44]. 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 [4] 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 Reasoning

Automated reasoning has traditionally focused on classical first-order logic but it is increasingly important for automation to other logics. We are applying our research to the following extensions to this traditional focus.

- 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 [38] continue to be the fastest automated theorem provers for propositional and first-order linear logic.
- 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 polarized and focused logic. One of the advantages of this approach is that it allows combining and extending such techniques. For example, the PSI project has applied this approach to proof to the problem of SAT-modulo-Theory. In that domain, logical reasoning is combined with domain-specific decision procedures. The PSI project has shown how to incorporate the call to decision procedures in the proof-theoretical framework of focused sequent calculi and the proof-search mechanisms that are related to it.

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. There is also the POPLMark challenge [37] which 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 been applying their research results to design and building systems to directly aid in both of these challenges. One important requirements for reasoning about 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. Also, 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 5.2).

4.3. Proof Certificates

Within the context of the ProofCert project, various members of the team have been building a flexible framework for the definition of the semantics of proof evidence. The emphasis is to attempt to capture as many forms of proof evidence as is possible. Using this framework, we have defined the semantics of all the following forms of proof evidence: natural deduction, expansion trees, matings, proof nets, resolution refutations, and Frege proofs. Given our framework, there is one kernel that can check all of these different forms of proof. Thus, one only needs to trust this one kernel in order to trust the output of a very wide range of theorem provers working in either intuitionistic or classical logics (see [20], [19], and [32].

5. Software and Platforms

5.1. Abella

Participants: Kaustuv Chaudhuri [correspondant], Matteo Cimini, Dale Miller, Olivier Savary-Bélanger, Yuting Wang.

Main web-site: http://abella-prover.org.

Abella is an interactive theorem prover based on the two-level logic approach. It consists of a sophisticated reasoning logic that supports induction, co-induction, and generic reasoning, and a specification logic that is based on logic programming. Abella was initially designed to reason about simple second-order Lambda Prolog programs, which is sufficient for the computational specifications.

During 2013, as part of the RAPT Associated Team, Chaudhuri and Yuting Wang (former intern from Univ. Minnesota) released version 2.0 of Abella, a culmination of nearly two years of work and a significant improvement in its expressivity. Specifically,

The Abella specification logic now supports the full higher-order hereditary Harrop logic of λProlog. This logic allows for very natural specifications of higher-order relations, and leads to cleaner and simpler proofs.

The Abella reasoning logic was extended with support for arbitrary dynamic contexts and incremental backchaining. The design is based on fundamental insights from *focusing*, a core strength of the team.

A number of illustrative examples of the use of higher-order reasoning were added to the Abella examples library, including a novel new characterization of marked β-reduction in the λ-calculus in terms of a simple higher-order inductive definition of λ-paths.

These results were published in PPDP 2013 [26].

Abella continues to evolve as part of RAPT. In 2013, we hosted an intern from McGill University, Olivier Savary-Bélanger (supervised by Chaudhuri), who investigated extensions of Abella with *regular context schemas*. Among his contributions:

Abella's reasoning level has been augmented with a *plugin* system that both extends the syntax of Abella theories and adds new tactics.

The main plugin for context schemas allows definitions of regular contexts and context relations, with entirely automatic proofs of the main administrative lemmas.

Experimentally, this extension can be used to eliminate up to 40% of the proof text, including nearly 100% of the administrative lemmas on contexts, from typical examples from the meta-theory of the λ-calculus.

We expect this extension to become part of the 2.1 release of Abella, scheduled for later in 2014.

One important application of Abella emerged in 2013: the formalization of bisimulation-up-to techniques for process calculi such as CCS and the π-calculus. Chaudhuri, Cimini, and Miller have formulated the correctness proof of a number of prominent up-to-techniques using the co-inductive and higher-order facilities of Abella. This work indicates an important emerging direction for Abella: modular reasoning.

In terms of development, we have welcomed Savary-Bélanger into the development team, and added a number of collaborators into the management team for the Abella web-site.

5.2. Bedwyr

Participants: Quentin Heath, Dale Miller [correspondant].

Main web-site: http://slimmer.gforge.inria.fr/bedwyr/.

During the first half of 2013, Quentin Heath was working as an engineer on the team, supported by the BATT ADJ project funded by Inria. During that time, we worked exclusively on making improvements to Bedwyr. In particular, he made extensive and important changes to the tabling mechanism of Bedwyr, a feature of model checking systems that is capable of remembering past successful proofs (it can even support a finite failure as a successful proof of a negation). These extension allow lemmas to be used to greatly extend the scope of what can be inferred from a table. For example, if we are attempting to show that there is a winning strategy for a given board position, we would certainly like to make use of a lemma that allows one to infer that winning strategies are preserved under symmetries of the board. There are a number of design issues that go along with the design of such a tabling mechanism: for example, should one use such lemmas in a forward-chaining or backward-chaining fashion. Quentin has tested both of these options in order to collect information as to what the trade-offs would be.

We should note that Quentin Heath is now a PhD student on the team and is addressing a number of theoretical questions related to his research on Bedwyr.

5.3. Profound

Participant: Kaustuv Chaudhuri [correspondant].

Profound is a new interactive theorem proving and proof-exploration tool based on the idea of building formal proofs *without* the use of formal proof languages. The core concepts are a generalization of *deep inference* for the underlying logical formalism, and *proof-by-pointing* for the user-interaction metaphors.

A user proves a theorem in *Profound* by using the keyboard and mouse to select subformulas of the theorem and dragging them to their suitable “destination”. For instance, the formula $(A \to C) \to (A \wedge B \to C)$ is proved by dragging the two As and the two Cs to each other. This kind of *direct manipulation* is nevertheless constrained by the system to be both correct—meaning that no manipulation is logically unsound—and complete—meaning that every provable theorem can be proved using these metaphors.

The system is still in its early stages, but it currently supports first-order classical linear logic. It has been documented in a paper at ITP 2013 [18].

We are in the process of extending the system to intuitionistic logics, and adding a back-end exporter for more traditional proof systems with formal proof languages such as Coq and Isabelle.

5.4. Psyche

Participants: Mahfuza Farooque, Stéphane Graham-Lengrand [correspondant].

Psyche (*Proof-Search factorY for Collaborative HEuristics*) is a modular proof-search engine whose first version, 1.0, was released in 2012:
http://www.lix.polytechnique.fr/~lengrand/Psyche/

Its motivation is twofold:

On the one hand, prove some mathematics of the broadest range while making the most of problem-specific techniques; On the other hand, gain high confidence about the correctness of the proofs produced without having to rely on a proof-checker.

Psyche’s proof-search mechanism is simply the incremental construction of proof-trees in the polarized and focused sequent calculus. Its architecture organizes an interaction between a trusted universal kernel and smart plugins that are meant be efficient at solving certain kinds of problems:

The kernel contains the mechanisms for exploring the proof-search space in a sound and complete way, taking into account branching and backtracking. The output of Psyche comes from the (trusted) kernel and is therefore correct by construction. The plugins then drive the kernel by specifying how the branches of the search space should be explored, depending on the kind of problem that is being treated. The quality of the plugin is then measured by how fast it drives the kernel towards the final answer.

In 2013, major developments were achieved in Psyche, which now handles classical propositional logic *modulo a theory* such as linear arithmetic, equality with uninterpreted symbols, arrays, etc. It therefore works in the same logic as Sat-Modulo-Theories (SMT) solvers and the architecture to handle such theories is the main contribution of 2013, in particular with the integration of the *simplex* algorithm.

Thanks to a plugin that simulates the behavior of a SAT-solver (DPLL) [21], the new version of Psyche can now simulate the behavior of SMT-solvers.

A lot of features inspired by SAT-solvers have now been lifted to proof-search in general, such as a *memoization table* to record and re-use known proofs, the technique of *2-watched literals* to efficiently propagate direct consequences of new hypotheses, machine learning techniques for *restart policies*, etc.

Psyche has been the topic of the 2013 publication [23].

6. New Results

6.1. Substitution as Proof Compression

Participants: Lutz Straßburger, Novak Novakovic.

In previous work [58] we have shown how the calculus of structures can accommodate Tseitin extension without relying on the cut (or modus ponens). Thus, cut and extension can be studied independently as proof compression mechanisms. Another such proof compression mechanism is substitution. It has been shown by Cook, Reckhow, Krajíček and Pudlák that in the presence of cut, extension and substitution are equally powerful with respect to proof complexity. This year we succeeded in showing that this is also the case in the absence of cut. I.e., we have shown that the cut-free system with extension and the cut-free system with substitution p-simulate each other. This result is presented in [34].

6.2. Herbrand Confluence

Participants: Lutz Straßburger, Stefan Hetzl.

In the result on Herbrand confluence from last year [46], the endsequent of a proof had to be an existential sentence in prenex form. This year we were able to relax this restriction and to extend our result to arbitrary endsequent. This work has been published in [15].

6.3. Nested Sequents for Intuitionistic Modal Logics

Participant: Lutz Straßburger.

We present cut-free deductive systems without labels for the intuitionistic variants of the modal logics obtained by extending IK with a subset of the axioms d, t, b, 4, and 5. For this, we use the formalism of nested sequents, which allows us to give a uniform cut elimination argument for all 15 logic in the intuitionistic S5 cube. This work (published in [25]), is an improvement of the result on intuitionistic modal logic from 2011: the deductive systems the cut elimination proof are much simpler now.

6.4. First efforts at designing proof certificates

Participants: Hichem Chihani, Quentin Heath, Dale Miller, Fabien Renaud.

Work on the ERC Advance Grant ProofCert has progressed along two lines.

Given earlier work within the team [6], [7], there now exists a flexible and well understood concept of focused proof for classical and intuitionistic first-order logics. Chihani, Miller, and Renaud have been working to use that notion of proof as a means of providing flexible definition of *proof evidence* for those two logics. Initial results along those directions have been reported in the [19] and [20]. In those papers, several examples definitions of the semantics of *proof certificates* (formal documents providing the details of some proof evidence) are provided in such a way that a single, simple proof checker can formally elaborate that evidence into a focused sequent calculus. Such an elaboration thus guarantees the soundness of that proof. These papers also describe a “reference proof checker” that has been built with the expectation that its formal correctness can be established. That checker is also able to do bounded *proof reconstruction* as well as allow both deterministic and non-deterministic computation to be mixed with deduction.

Our understanding of focused proofs in the presence of both induction and co-induction (inference rules found in model checkers and most theorem provers) is less well developed. As a result, Miller and Tiu have been studied a simple approach of proof certificate in the setting of model checking in the hope of identifying the relevant proof theory designs that need to be developed. In [33], they showed how tabled deduction in model checking can be used to provide a formal proof certificate for a range of co-inductively defined predicates.

6.5. Combinations of classical and intuitionistic logic

Participant: Dale Miller.

Chuck Liang and Miller have been studying the question of how one can mix intuitionistic and classical logic into a single logic. The initial motivation for considering this problem arose from the concerns raised by the ProofCert project of how best to deal with both classical and intuitionistic logic and their associated proof evidence. Will there need to be two different kinds of checkers and two different kinds of libraries for these two different kinds of logics? Will we be able to mix theorems and proofs in one logic with those in the second logic in rich and useful ways?

One way we have considered answering this question is to actually consider a third logic that combines these other two logics. Our work on such combinations is reported in [16], where a thorough analysis of the semantics and proof theory of such a combination is provides, and in [24], where significant examples of the computational aspects of proofs are explored in detail.

6.6. Formal meta theory of sequent calculus

Participant: Dale Miller.

Keeping with the ProofCert theme of finding global, eternal, and formal mechanisms representing proof evidence, Miller and Pimentel describe in [17] a way in which linear logic can be used to formally specify inference rules for a wide range of proof system in several logics. They were able to show that adequacy of their encodings and to provide sufficient conditions for both cut-elimination and initial-elimination to hold for the resulting proof systems. The fact that these elimination results hold or not is an important characteristic for judging a proof system. Using this work, these important questions can be resulted automatically for a wide range of such proof systems.

6.7. The correctness of program using finite precision

Participants: Ivan Gazeau, Dale Miller.

Programs dealing with real number quantities must live with the fact that such numbers are represented using only finite precision. As such, programs that might be considered correct over the abstract field of infinite precision arithmetic can display chaotic and incorrect behaviors when run on actual computer hardware.

One such problem with finite precision is that programs can “leak” information about values that are intended to be hidden or at least obfuscated as happens in the area differential privacy. In [22], Gazeau, Miller, and Palamidessi illustrated just how such attacks on information hiding can be made and how it is possible to add noise to reported data values in such a way that only appropriate amounts of information leakage occurs.

In his PhD thesis, *Safe Programming in finite precision: Controlling the errors and information leaks* (École Polytechnique, 2013 [11]), Gazeau develops that theme further as well as shows how techniques from rewriting theory can be applied to show that, in some situations, the chaotic behavior of finite precision programs can be expected to converge in acceptable time to acceptable answers.

6.8. Sequent Calculus with Calls to a Decision Procedure

Participants: Mahfuza Farooque, Stéphane Graham-Lengrand.

In the PSI project, a version of the focused sequent calculus (for first-order classical logic) has been designed, which can call external decision procedures. Several results were achieved in 2013 since the last Activity Report:

Firstly, a bug was discovered in the proof of cut-elimination, which was used to prove the logical completeness of the calculus. Fixing the bug required minor changes in the definition of the system, but incurred a major re-development of the meta-theory. Out of this technical work, one idea emerged: in presence of a non-trivial theory, changing the polarity of literals may change the provability of formulas. This was quite unexpected, but it led to interesting issues, such as finding sufficient conditions on polarities to guarantee cut-elimination and logical completeness. An substantial achievement in this research topic was to successfully address such issues, which gave rise to a new version of the report [30].

Secondly, more techniques from automated reasoning were captured as proof-search in this sequent calculus (the incremental construction of proof-trees): besides the SMT-solving algorithm DPLL(T) treated successfully in 2012 (which was written down and published this year in [21]), the techniques of *clause tableaux* and *connection tableaux* were captured this year. This includes in particular a notion of *clause tableaux modulo theories* that C. Tinelli introduced in 2007 [60]. This new range of captured techniques is interesting as clause tableaux are designed to handle quantifiers, which DPLL(T) does not. This gives a new hope to combine the efficiency of SAT-solvers for propositional reasoning with the handling of quantifiers.

6.9. Path Functors in the Category of Small Categories

Participant: François Lamarche.

In [31] François Lamarche gives a detailed description of two path functors in the category of small categories, which he calls $\mathbf{Pe}$ and $\mathbf{P}$, and proves some of their important properties. The second of these is the functor which is used to model the Martin-Löf identity type in [47]; it associates to every small category X an internal category structure whose object of objects is X; one important theorem which is proved in [31] is that the category of internal (co- or contravariant) presheaves on $\mathbf{P}X$ coincides with the category of Grothendieck bifibrations over the base X. Thus, through a trivial use of monadic abstract nonsense, we can say that $\mathbf{P}X$ is the free bifribration over X. The category $\mathbf{P}X$ is obtained by taking the bigger $\mathbf{Pe}X$, which is a little more than just a category, being poset-enriched, and getting rid of the order enrichment by quotienting. $\mathbf{Pe}X$ is a more general kind of bifibration than an ordinary Grothendieck bifibration, and the enrichment is necessary to describe its properties, thus taking us outside of the theory 1-categores.

6.10. Subformula Linking as an Interaction Method

Participant: Kaustuv Chaudhuri.

We showed how to generalize the *calculus of structures*, a *deep inference* formalism, for classical linear logic to a *calculus of linking* [18]. This generalization simplifies the calculus by eliminating most of its inference rules. In its place we add a notion of annotation with *links* and a *link resolution* procedure. We show that this is sound and complete with respect to the usual calculus of structures. The linking calculus is the foundational basis of the *Profound* tool described in 5.1.

6.11. Recovering Proof Structures in the Sequent Calculus

Participants: Kaustuv Chaudhuri, Stefan Hetzl, Dale Miller.

The *sequent calculus* is often criticized as a proof syntax because it contains a lot of noise. It records the precise minute sequence of operations that was used to construct a proof, even when the order of some proof steps in the sequence is irrelevant and when some of the steps are unnecessary or involve detours. These features lead to several technical problems: for example, cut-elimination in the classical sequent calculus LK, as originally developed by Gentzen, is not confluent, and hence proof composition in LK is not associative. Many people choose to discard the sequent calculus when attempting to design a better proof syntax with the desired properties.

In recent years, there has been a project at Parsifal to recover some of these alternative proof syntaxes by imposing a certain abstraction over sequent proofs. Our technique, pioneered at Parsifal, involves the use of *maximal multi-focusing* which gives a syntactic characterization of those sequent proofs that: (1) have a "don't care" ordering of proof steps where the order does not matter, and (2) groups larger logical steps, called *actions*, into a maximally parallel form where only important orderings of actions are recorded. The earliest example of this technique was in [40], where we showed a class of sequent proofs that were isomorphic to proof nets for multiplicative linear logic. In 2012, we were able to obtain a similar result for first-order classical logic, wherein we defined a class of sequent proofs that are isomorphic to expansion proofs, a generalization of Herbrand disjunctions that is in some sense a minimalistic notion of proof for classical logic. This result was published in a preliminary form at the CSL 2012 conference [39].

In 2013 we published an extended paper on this result in the Journal of Logic and Computation [14]. The major contribution here was a detailed proof of the result that gives a precise account of the proof identifications made by expansion proofs.

7. Partnerships and Cooperations

7.1. European Initiatives

7.1.1. FP7 Projects

7.1.1.1. Proofcert

Participants: Hichem Chihani, Quentin Heath, Dale Miller [correspondant], Fabien Renaud.

Title: ProofCert: Broad Spectrum Proof Certificates

Duration: January 2012 - December 2016

Type: IDEAS

Instrument: ERC Advanced Grant

Coordinator: Dale Miller

Abstract: There is little hope that the world will know secure software if we cannot make greater strides in the practice of formal methods: hardware and software devices with errors are routinely turned against their users. The ProofCert proposal aims at building a foundation that will allow a broad spectrum of formal methods—ranging from automatic model checkers to interactive theorem provers—to work together to establish formal properties of computer systems. This project starts with a wonderful gift to us from decades of work by logicians and proof theorist: their efforts on logic and proof has given us a *universally accepted* means of communicating proofs between people and computer systems. Logic can be used to state desirable security and correctness properties of software and hardware systems and proofs are uncontroversial evidence that statements are, in fact, true. The current state-of-the-art of formal methods used in academics and industry shows, however, that the notion of logic and proof is severely fractured: there is little or no communication between any two such systems. Thus any efforts on computer system correctness is needlessly repeated many time in the many different systems: sometimes this work is even redone when a given prover is upgraded. In ProofCert, we will build on the bedrock of decades of research into logic and proof theory the notion of *proof certificates*. Such certificates will allow for a complete reshaping of the way that formal methods are employed. Given the infrastructure and tools envisioned in this proposal, the world of formal methods will become as dynamic and responsive as the world of computer viruses and hackers has become.

7.1.2. Collaborations in European Programs, except FP7

7.1.2.1. STRUCTURAL: ANR blanc International

Participants: Kaustuv Chaudhuri, Nicolas Guenot, Willem Heijltjes, Stefan Hetzl, Novak Novakovic, François Lamarche, 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, who are all 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. International Initiatives

7.2.1. Inria Associate Teams

7.2.1.1. RAPT

Participants: Kaustuv Chaudhuri [correspondant], Dale Miller, Yuting Wang, Olivier Savary-Bélanger.

Title: Applying Recent Advances in Proof Theory for Specification and Reasoning

Inria principal investigator: Kaustuv Chaudhuri

International Partner:

Institution: McGill University (Canada)

Laboratory: School of Computer Science

Researcher: Prof. Brigitte Pientka

International Partner:

Institution: University of Minnesota (United States)

Laboratory: Department of Computer Science and Engineering

Researcher: Prof. Gopalan Nadathur

International Partner:

Institution: Carnegie Mellon University (United States)

Laboratory: Department of Computer Science

Researcher: Prof. 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.

7.2.2. Inria International Partners

7.2.2.1. PHC Procope: From Proofs to Counterexamples for Programming

Participants: Kaustuv Chaudhuri, Nicolas Guenot, Willem Heijltjes, Lutz Straßburger.

Title: From Proofs to Counterexamples for Programming

Duration: 01/01/2012 – 31/12/2013

German Partner: University of Bonn, Institute for Computer Science (Department III)

Finding counterexamples is an endeavor which is as important as proving theorems. But while the latter has seen a huge amount of research effort—we have nowadays a large quantity of tools for automated and interactive theorem proving—the former has mainly been neglected by proof theorists. One of the reasons is that finding counterexamples or countermodels has been considered a model theoretical activity, rather than a proof theoretical one. Only recently, researchers have begun to explore the well-known duality between "proof search" and "search for countermodels" in a purely proof theoretical way. The main objective of this collaboration is to develop the necessary proof theory for automatically generating such counterexamples in a more general setting.

7.3. International Research Visitors

7.3.1. Visits of International Scientists

Chuck Liang (Professor from Hofstra University, NY, USA) visited for three weeks in May and June and another week in December.

Gopalan Nadathur (Professor from the University of Minnesota) visited for two weeks in May and June.

Elaine Pimentel (Associate Professor, UFRN, Brazil) for four weeks in June and July.

7.3.2. Internships

Olivier Savary-Bélanger (Masters, McGill University, Canada), supervised by Kaustuv Chaudhuri

7.3.3. Visits to International Teams

Fabien Renaud visited Gopalan Nadathur in Minneapolis for two weeks in February.

Dale Miller visited Alwen Tiu at the Australian National University in Canberra, Australia for onc week in May 2013.

Dale Miller visited Christof Benzmüller for one week in February.

8. Dissemination

8.1. Scientific Animation

8.1.1. Organization

Dale Miller served on the Program Committee of Tableaux 2013, 16-19 September, Nancy, France.

Dale Miller is the editor-in-chief of the ACM Transactions on Computational Logic (ToCL) (June 2009 - May 2015).

Dale Miller has editorial duties on the following three other journals: *Journal of Automated Reasoning*, published by Springer (member of Editorial Board since 2011), *Theory and Practice of Logic Programming* published by Cambridge University Press (an editorial advisor since 1999), and *Journal of Applied Logic*, published by Elsevier (an area editor for "Type Theory for Theorem Proving Systems" since 2003).

Dale Miller is a member of the selection jury for the 2013 E. W. Beth Dissertation Award of the Association for Logic, Language and Information.

8.1.2. Invited Talks

François Lamarche gave a total of eight hours of lectures on path functors at the Groupe de Travail sur les Catégories supérieures, polygraphes et homotopie at the PPS laboratory, Université Paris VII.

Dale Miller gave an invited talk at LFMTP 2013: Logical Frameworks and Meta-Languages: Theory and Practice, affiliated with ICFP'13, Boston, 23 September 2013.

Dale Miller is on the Advisory Board for LICS (for 2012 - 2015) and is a member of the Steering Committee of CPP since 2012.

Dale Miller gave invited departmental colloquia at the College of Engineering and Computer Science, Australian National University, 14 May and the Department of Mathematics and Computer Science, Freie Universität Berlin, 22 February.

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

Licence: Stéphane Graham-Lengrand teaches 50 hours (eq. TD) in L3 at Ecole Polytechnique in the course “INF431: Algorithmique et programmation”.

Master: Stéphane Graham-Lengrand teaches 36 hours (eq. TD) in M1 at Ecole Polytechnique in the course “INF551: Computer-aided reasoning”, and 15 hours (eq. TD) in M2 at Master Parisien de Recherche en Informatique (MPRI) on “Curry-howard correspondence for classical logic”.

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.

Dale Miller was an invited lecturer at the CUSO Winter School in Mathematics and Computer Science “Proof and Computation”, Les Diablerets, Switzerland, 27-31 January 2013.

8.2.2. Supervision

PhD: Nicolas Guenot, “Nested Deduction in Logical Foundations for Computation”, Ecole Polytechnique, April 10 2013, supervisor Lutz Straßburger (thesis available at [12])

PhD: Ivan Gazeau, “Safe Programming in finite precision: Controlling the errors and information leaks”, Ecole Polytechnique, October 14 2013, supervisors: Dale Miller and Catuscia Palamidessi [11].

PhD: Mahfuza Farooque, “Automated reasoning techniques as proof search in sequent calculus”, Ecole Polytechnique, December 19 2013, supervisor: Stéphane Graham-Lengrand [29].

PhD in progress: Quentin Heath, since October 2013, supervisor: Dale Miller.

PhD in progress: Zakaria Chihani, since October 2012, supervisor: Dale Miller.

PhD in progress: Hernán Vanzetto, since October 2010, co-supervisors: Stefan Merz and Kaustuv Chaudhuri.

8.2.3. Juries

François Lamarche was member of the PhD jury of Pierre Rannou, Université Aix-Marseille, October 21 2013.

Dale Miller was a member of the PhD jury of the following three students during 2013: Mahfuza Farooque, Ecole Polytechnique 19 December 2013 (evaluator); Stéphane Zimmermann, University of Paris Diderot, 10 December 2013 (president); Matthias Puech, University of Bologna, 8 April 2013 (reporter).

Stéphane Graham-Lengrand was member of the PhD jury of Sophia Knight, École Polytechnique, September 20 2013.

9. Bibliography

Major publications by the team in recent years

[1] 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 für Informatik, September 2011, pp. 159–173, http://drops.dagstuhl.de/opus/frontdoor.php?source_opus=3229

[2] A. Gacek, D. Miller, G. Nadathur. *Nominal abstraction*, in "Information and Computation", 2011, vol. 209, n^{o} 1, pp. 48–73, http://arxiv.org/abs/0908.1390

[3] 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, pp. 284–293 [*DOI :* 10.1109/LICS.2010.12], http://www.lix.polytechnique.fr/~lutz/papers/AFII.pdf

[4] 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, pp. 246–261

[5] STÉPHANE. LENGRAND, R. DYCKHOFF, J. MCKINNA. *A Focused Sequent Calculus Framework for Proof Search in Pure Type Systems*, in "Logical Methods in Computer Science", 2011, vol. 7, n^{o} 1, http://www.lix.polytechnique.fr/~lengrand/Work/Reports/TTSC09.pdf

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

[7] C. LIANG, D. MILLER. *A Focused Approach to Combining Logics*, in "Annals of Pure and Appl. Logic", 2011, vol. 162, n^{o} 9, pp. 679–697 [*DOI :* 10.1016/J.APAL.2011.01.012], http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/lku.pdf

[8] 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), LNCS, 2011, vol. 7086, pp. 54–69, http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/cpp11.pdf

[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, pp. 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] I. GAZEAU. , *Programmation sûre en précision finie : Contrôler les erreurs et les fuites d'informations*, Ecole Polytechnique X, October 2013, http://hal.inria.fr/pastel-00913469

[12] N. GUENOT. , *Nested Deduction in Logical Foundations for Computation*, Ecole Polytechnique, April 2013, http://www.itu.dk/people/ngue/pub/thesis.pdf

Articles in International Peer-Reviewed Journals

[13] A. BERNADET, S. GRAHAM-LENGRAND. *Non-idempotent intersection types and strong normalisation*, in "Logical Methods in Computer Science", 2013, vol. 9, n^{o} 4, pp. 17-42, http://hal.inria.fr/hal-00906778

[14] K. CHAUDHURI, S. HETZL, D. MILLER. *A Multi-Focused Proof System Isomorphic to Expansion Proofs*, in "Journal of Logic and Computation", 2014, http://hal.inria.fr/hal-00937056

[15] S. HETZL, L. STRASSBURGER. *Herbrand-Confluence*, in "LMCS", 2013, vol. 9, n^{o} 4:24, 25 p. , final version, accepted for publication [*DOI :* 10.2168/LMCS-9(4:24)2013], http://hal.inria.fr/hal-00925641

[16] C. LIANG, D. MILLER. *Kripke Semantics and Proof Systems for Combining Intuitionistic Logic and Classical Logic*, in "Annals of Pure and Applied Logic", February 2013, vol. 164, n^{o} 2, pp. 86-111 [*DOI :* 10.1016/J.APAL.2012.09.005], http://hal.inria.fr/hal-00787601

[17] D. MILLER, E. PIMENTEL. *A formal framework for specifying sequent calculus proof systems*, in "Theoretical Computer Science", February 2013, pp. 98-116 [*DOI :* 10.1016/J.TCS.2012.12.008], http://hal.inria.fr/hal-00787586

International Conferences with Proceedings

[18] K. CHAUDHURI. *Subformula Linking as an Interaction Method*, in "4th Conference on Interactive Theorem Proving", Rennes, France, Lecture Notes in Computer Science, Springer, July 2013, vol. 7998, pp. 386-401 [*DOI :* 10.1007/978-3-642-39634-2_28], http://hal.inria.fr/hal-00937009

[19] Z. CHIHANI, D. MILLER, F. RENAUD. *Checking foundational proof certificates for first-order logic*, in "PxTP - Proof Exchange for Theorem Proving", Lake Placid, United States, June 2013, http://hal.inria.fr/hal-00906486

[20] Z. CHIHANI, D. MILLER, F. RENAUD. *Foundational proof certificates in first-order logic*, in "CADE - 24th International Conference on Automated Deduction", Lake Placid, United States, June 2013, http://hal.inria.fr/hal-00906485

[21] M. FAROOQUE, S. LENGRAND, A. MAHBOUBI. *A bisimulation between DPLL(T) and a proof-search strategy for the focused sequent calculus*, in "LFMTP - International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice - 2013", Boston, United States, A. MOMIGLIANO, B. PIENTKA, R. POLLACK (editors), ACM, September 2013 [*DOI :* 10.1145/2503887.2503892], http://hal.inria.fr/hal-00854426

[22] I. GAZEAU, D. MILLER, C. PALAMIDESSI. *Preserving differential privacy under finite-precision semantics*, in "QAPL - 11th International Workshop on Quantitative Aspects of Programming Languages and Systems", Rome, Italy, L. BORTOLUSSI, H. WIKLICKY (editors), Electronic Proceedings in Theoretical Computer Science, Open Publishing Association, 2013, vol. 117, pp. 1-18 [*DOI :* 10.4204/EPTCS.117.1], http://hal.inria.fr/hal-00780774

[23] S. GRAHAM-LENGRAND. *Psyche: a proof-search engine based on sequent calculus with an LCF-style architecture*, in "22nd International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (Tableaux'13)", Nancy, France, D. GALMICHE, D. LARCHEY-WENDLING (editors), Lecture Notes in Computer Science, Springer-Verlag, 2013, vol. 8123, pp. 149–156, http://hal.inria.fr/hal-00906789

[24] C. LIANG, D. MILLER. *Unifying Classical and Intuitionistic Logics for Computational Control*, in "Logic in computer science (LICS 2013)", New Orleans, United States, O. KUPFERMAN (editor), June 2013, http://hal.inria.fr/hal-00906299

[25] L. STRASSBURGER. *Cut Elimination in Nested Sequents for Intuitionistic Modal Logics*, in "FoSSaCS 2013", Rome, Italy, F. PFENNING (editor), LNCS, Springer, March 2013, vol. 7794, pp. 209-224 [*DOI :* 10.1007/978-3-642-37075-5_14], http://hal.inria.fr/hal-00770678

[26] Y. WANG, K. CHAUDHURI, A. GACEK, G. NADATHUR. *Reasoning About Higher-Order Relational Specifications*, in "International Symposium on Principles and Practice of Declarative Programming", Madrid, Spain, T. SCHRIJVERS (editor), ACM, September 2013 [*DOI :* 10.1145/2505879.2505889], http://hal.inria.fr/hal-00787126

Scientific Books (or Scientific Book chapters)

[27] , *Proceedings of the Sixth Workshop on Intersection Types and Related Systems (ITRS'12)*, EPTCS, 2013, vol. 121, pp. 1–93 [*DOI :* 10.4204/EPTCS.121], http://hal.inria.fr/hal-00912611

Research Reports

[28] K. CHAUDHURI, J. DESPEYROUX. , *A Hybrid Linear Logic for Constrained Transition Systems with Applications to Molecular Biology*, October 2013, 30 p. , http://hal.inria.fr/inria-00402942

[29] M. FAROOQUE. , *Automated reasoning techniques as proof-search in sequent calculus*, October 2013, Version of thesis at time of defense, http://hal.inria.fr/hal-00939124

[30] M. FAROOQUE, S. GRAHAM-LENGRAND. , *Sequent Calculi with procedure calls*, September 2013, http://hal.inria.fr/hal-00779199

Other Publications

[31] F. LAMARCHE. , *Path Functors in Cat*, September 2013, http://hal.inria.fr/hal-00831430

[32] D. MILLER. , *Communicating and trusting proofs: The case for foundational proof certificates*, January 2013, To appear in the Proceedings of the 14th Congress of Logic, Methodology and Philosophy of Science in Nancy, France, 19-26 July 2011, http://hal.inria.fr/hal-00772727

[33] D. MILLER, A. TIU. , *Extracting Proofs from Tabled Proof Search*, September 2013, http://hal.inria.fr/hal-00863561

[34] L. STRASSBURGER, N. NOVAKOVIC. , *On the Power of Substitution in the Calculus of Structures*, November 2013, submitted, http://hal.inria.fr/hal-00925707

References in notes

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

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

[37] 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, pp. 50–65

[38] 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

[39] K. CHAUDHURI, S. HETZL, D. MILLER. *A Systematic Approach to Canonicity in the Classical Sequent Calculus*, in "Computer Science Logic (CSL'12): 21st Annual Conference of the EACSL", P. CÉGIELSKI, A. DURAND (editors), Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-

Zentrum für Informatik, September 2012, vol. 16, pp. 183–197, http://drops.dagstuhl.de/opus/frontdoor.php?source_opus=3672

[40] 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, pp. 383–396, http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/tcs08trackb.pdf

[41] 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, pp. 33–44, http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/lics08a.pdf

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

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

[44] 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, pp. 1–36, http://arxiv.org/abs/0709.1205

[45] 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, pp. 54–68

[46] S. HETZL, L. STRASSBURGER. *Herbrand-Confluence for Cut-Elimination in Classical First-Order Logic*, in "Computer Science Logic (CSL) 2012", P. CÉGIELSKI, A. DURAND (editors), Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012, vol. 16, pp. 320–334

[47] F. LAMARCHE. *Modeling Martin Löf Type Theory in Categories*, in "Logic, Categories, Semantics", Bordeaux, France, C. RETORÉ, J. GILIBERT (editors), Elsevier North-Holland, 2014, vol. 12, special Issue of J. of Applied Logic [*DOI :* 10.1016/J.JAL.2013.08.003], http://hal.inria.fr/hal-00706562

[48] P. MARTIN-LÖF. *Constructive Mathematics and Computer Programming*, in "Sixth International Congress for Logic, Methodology, and Philosophy of Science", Amsterdam, North-Holland, 1982, pp. 153–175

[49] 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, pp. 80–136, http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/mcdowell01.pdf

[50] 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, pp. 434–445

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

[52] D. MILLER. *Forum: A Multiple-Conclusion Specification Logic*, in "Theoretical Computer Science", September 1996, vol. 165, n^O 1, pp. 201–232

[53] 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, pp. 125–157

[54] 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, pp. 118–127, http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/lics03.pdf

[55] 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, pp. 202–206

[56] 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, pp. 15–21

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

[58] L. STRASSBURGER. *Extension without Cut*, in "Annals of Pure and Applied Logic", 2012, vol. 163, n^O 12, pp. 1995–2007 [*DOI :* 10.1016/J.APAL.2012.07.004], http://hal.inria.fr/hal-00759215

[59] THE COQ DEVELOPMENT TEAM. , *The Coq Proof Assistant Version 8.3 Reference Manual*, Inria, October 2010

[60] C. TINELLI. *An Abstract Framework for Satisfiability Modulo Theories*, in "Proceedings of the 16th International Conference on Tableaux methods (Tableaux'07)", N. OLIVETTI (editor), LNCS, Springer-Verlag, 2007, vol. 4548, http://dl.acm.org/citation.cfm?id=1420269.1420273

[61] 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