IN PARTNERSHIP WITH:
CNRS

Ecole Polytechnique

Activity Report 2014

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

PhD Students

Roberto Blanco Martinez [Inria, from Oct 2014]
Hichem Chihani [Inria, from Oct 2012]
Quentin Heath [Ecole Polytechnique, from Sep 2013]
Sonia Marin [Inria, from Nov 2014]

Post-Doctoral Fellows

Ryuta Arisaka [Inria, until May 2014]
Taus Brock-Nannestad [Inria, from Oct 2014, granted by FP7 ERC PROOFCERT- project]
Matteo Cimini [Inria, until Jan 2014, granted by FP7 ERC PROOFCERT- project]
Anupam Das [Inria, Nov 2013 – Sep 2014]
Danko Ilik [Inria, granted by FP7 ERC PROOFCERT- project]
Giselle Machado Nogueira Reis [Inria, from Nov 2014, granted by FP7 ERC PROOFCERT- project]
Fabien Renaud [Inria, until Sep 2014, granted by FP7 ERC PROOFCERT- project]
Marco Volpe [Inria, from Nov 2014, granted by ERCIM and FP7 ERC PROOFCERT- project]

Visiting Scientists

Roman Kuznets [Vienna University of Technology, funded through Digiteo, 24 Nov – 5 Dec 2014]
Chuck Liang [Professor, Hofstra University, USA, from 20 May – 7 Jun 2014]
Gopalan Nadathur [Professor, University of Minnesota, 20 May – 2 Jun 2014]
Claudio Sacerdoti Coen [University of Bologna, from 29 Sep – 3 Oct 2014]
Yuting Wang [University of Minnesota, funded through ERC ProofCert, May – Aug 2014]

Administrative Assistant

Martine Thirion [Inria]

Others

Siddhartha Prasad [Inria, Intern, from Jun 2014 until Jul 2014]
Mary Southern [University of Minnesota, Intern, funded through ERC ProofCert, May – Aug 2014]

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.

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 [64], and normalization in different logics can justify the design of new and different functional programming languages [38].
- *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 [70], and different proof search strategies can be used to justify the design of new and different logic programming languages [68].

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 [39] 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 [76] 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 [72] 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 [73], 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. Inductive and co-inductive reasoning

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 is based on earlier work by McDowell, Miller, and Tiu [66] [65] [71] [77], and on more recent work by Gacek, Miller, and Nadathur [3] [52]. The Parsifal team, along with our colleagues in Minneapolis, Canberra, Singapore, and Cachen, have been building two tools that exploit the novel features of this logic. These two 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.

We have used these systems to provide formalize reasoning of a number of complex formal systems, ranging from programming languages to the λ-calculus and π-calculus.

During 2014, the Abella system has been extended with a number of new features. A number of new significant examples have been implemented in Abella and an extensive tutorial for it has been written [31].

3.3. Developing a foundational approach to defining proof evidence

The team is developing a framework for defining the semantics of proof evidence. With this framework, implementers of theorem provers can output proof evidence in a format of their choice: they will only need to be able to formally define that evidence's semantics. With such semantics provided, proof checkers can then check alleged proofs for correctness. Thus, anyone who needs to trust proofs from various provers can put their energies into designing trustworthy checkers that can execute the semantic specification.

In order to provide our framework with the flexibility that this ambitious plan requires, we have based our design on the most recent advances within the theory of proofs. For a number of years, various team members have been contributing to the design and theory of *focused proof systems* [40] [42] [44] [45] [55] [62] [63] and we have adopted such proof systems as the corner stone for our framework.

We have also been working for a number of years on the implementation of computational logic systems, involving, for example, both unification and backtracking search. As a result, we are also building an early and reference implementation of our semantic definitions.

3.4. Deep inference

Deep inference [57], [59] 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.

3.5. 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 [54] for linear logic but also in Robinson’s proof nets [74] 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., [5], [4] and [58]. 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 [5] 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. Integrating a model checker and a theorem prover

The goal of combining model checking with inductive and co-inductive theorem in a rather appealing one. The strengths of systems in these two different systems are strikingly different. A model checker is capable of exploring a finite space automatically: such a tool can repeatedly explores all possible cases for how a computational space can be explored. On the other hand, a theorem prover might be able to prove clever things about a search space. For example, a model checker could attempt to discover whether or not there exists a winning strategy for, say, tic-tac-toe while an inductive theorem prover might be able to prove that if there is a winning strategy from one board then there is a winning strategy from any symmetric version of that board. Of course, being about to combine proofs from these system could drastically reduce the state exploration and proof certificate that needs to be produced to prove the existence of winning strategies.

Our first step to providing an integration of model checking and (inductive) theorem proving was to develop a strong logic, we call $\mathcal{G}$, that extends intuitionistic logic with notions of least and greatest fixed points. We have developed the proof theory of this logic in earlier papers [3] [52]. We have now recently converted the Bedwyr system so that it formally accepts almost all definitions and statements of theorems that are accepted by the inductive theorem prover Abella. Thus, these two systems are proving theorems in the same logic and their theorems can now be shared.

The tabling mechanism of Bedwyr has been extended so that its it can make use of previously proved lemmas. Thus, when a goal to prove that some board position has a winning strategy, the lemma can to conclude yes if some symmetric board position is already in the table.

For more about recent progress on providing checkable proof certificates for model checking, see the web site for Bedwyr http://slimmer.gforge.inria.fr/bedwyr/.

4.2. Implementing trusted proof checkers

Traditionally, theorem provers—whether interactive or automatic—are usually monolithic: if any part of a formal development was to be done in a particular theorem prover, then all parts of it would need to be done in that prover. Increasingly, however, formal systems are being developed to integrate the results returned from several, independent and high-performing, specialized provers: see, for example, the integration of Isabelle with an SMT solver [51] as well as the Why3 and ESC/Java systems.

Within the Parsifal team, we have been working on foundational aspects of this problem of integrating different provers. As we have described above, we have been developing a formal framework for defining the semantics of proof evidence. We have also been working on building prototype checkers of proof evidence which are capable to executing such formal definitions. The proof definition language described in the papers [47], [46] is currently given an implementation in the λProlog programming language [69]. This initial implementation will be able to serve as a "reference" proof checker: others developing proof evidence definitions will be able to use this reference checker to make sure that they are getting their definitions to do what they expect.

Using λProlog as an implementation language has both good and bad points. The good points are that it is rather simple to confirm that the checker is, in fact, sound. The language also supports a rich set of abstracts which make it impossible to interfere with the code of the checker (no injection attacks are possible). On the negative side, however, the performance of our λProlog interpreters is lower than specially written checkers and kernels.

4.3. Trustworthy implementations of theorem proving techniques

Instead of integrating different provers by exchanging proof evidence and relying on a back-end proof-checker, another approach to integration consists in re-implementing the theorem proving techniques as proof-search strategies, on an architecture that guarantees correctness. Focused systems can serve as the basis of such

an architecture, identifying points of choice and backtrack and providing primitives for the exploration of the search space. These form a trusted *Application Programming Interface* that can be used to program and experiment various proof-search heuristics without worrying about correctness. No proof-checking is needed if one trusts the implementation of the API.

Following the description, in this framework, of quantifier-free techniques such as DPLL(T) [2], we are now exploring how the architecture can be adapted to accommodate techniques that handle quantifiers. In particular, unification-based or triggers-based techniques [37], [49].

This approach has led to the development of the Psyche engine.

5. New Software and Platforms

5.1. Abella

Participants: Kaustuv Chaudhuri [correspondant], Matteo Cimini [Indiana University], Dale Miller, Olivier Savary-Bélanger [Princeton University], Mary Southern [University of Minnesota], Yuting Wang [University of Minnesota].

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

Abella is an interactive theorem prover for reasoning about data structures with binding constructs using the λ-tree approach to syntax. It consists of a sophisticated reasoning logic that supports induction, co-induction, and generic reasoning. Abella also supports the *two-level logic approach* by means of a specification logic based on the logic programming language λProlog.

In 2014, the following additions were made to the system.

- A new translation layer was added to Abella's specification layer, which was used to build an interface to the LF dependent type theory [61]. This extension was documented in the following paper: [27]. A number of examples of the use of this new specification language are available at the following URL: http://abella-prover.org/lf
- Two minor releases were made, versions 2.0.2 and 2.0.3, that fixed a number of bugs and added several convenience features. Consult the change log for more details.

Accompanying these additions were the following publications.

- A new comprehensive tutorial for the Abella system has been accept to appear in the *Journal of Formalized Reasoning* [31].
- The new tactical plugin architecture and the dynamic contexts plugin of Abella in the following paper: [26].
- The use of co-induction and higher-order relations to formalize the meta-theory of various bisimulation-up-to techniques for common process calculi: [19].

5.2. Bedwyr

Participants: Quentin Heath, Dale Miller [correspondant].

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

Quentin Heath has continued to maintain and enhance this model checking system. In particular, the tabling mechanism has been extended and formalized to a greater extent. The tabling mechanism is now able to use Horn clause lemmas in order to increase the power of the table. For example, given this enhancement it is possible to tell Bedwyr that if a given board position (in some game) has a winning strategy then symmetric versions of that board also have winning strategies. Thus, when a given board position is recognized as winning, then table will understand that all symmetric versions of that board are winning.

Significant energies have also gone into trying to understand how cyclic proofs (recognized using the tabling mechanism) can be turned into certifiable proof evidence. Good results are currently developed for treating bisimulation and non-reachability: in these cases, cyclic proofs are used to supply invariants for induction and co-induction.

5.3. Psyche

Participants: Stéphane Graham-Lengrand [correspondant], Assia Mahboubi, Jean-Marc Notin.

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/

The engine implements the ideas developed in the section "Trustworthy implementations of theorem proving techniques" above, and was the object of the system description [56].

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 how fast it drives the kernel towards the final answer.

In 2014, major developments were achieved in Psyche, whose version 2.0 was released on 20th September 2014. It is now equipped with the machinery to handle quantifiers and quantifier-handling techniques. Concretely, it uses meta-variables to delay the instantiation of existential variables, and constraints on meta-variables are propagated through the various branches of the search-space, in a way that allows local backtracking. The kernel, of about 800 l.o.c., is purely functional.

6. New Results

6.1. Highlights of the Year

Dale Miller's 1994 LICS paper titled "A Multiple-Conclusion Meta-Logic" [67] was a co-recipient of the LICS Test of Time Award.

6.2. Modular Systems for Classical and Intuitionistic Logic

Participants: Sonia Marin, Lutz Straßburger.

Last year we have shown deductive systems for all intuitionistic modal logics in the modal S5-cube using logical rules in nested sequents [75]. This year we managed to exhibit fully modular systems. That is to say that there is a bijective correspondence between the modal axioms and the inference rules in the deductive system. This is achieved by using a combination of structural and logical rules. This result has been presented at AiML 2014 [24].

6.3. Nested Sequents for Constructive Modal Logics

Participants: Ryuta Arisaka, Anupam Das, Lutz Straßburger.

In the propositional case, "constructive" and "intuitionistic" logic are usually considered the same. However, in the presence of the modalities $\Box$ and $\Diamond$ this situation changes because there are several choice of which variants of the k-axiom (which are all equivalent in the classical case) are to be included. Whereas in [75] the intuitionistic variant of the S5-cube has been studied, we studied in this years work [34] the constructive variant of the logics in the S5-cube.

6.4. Intuitionistic Logic in the Calculus of Structures

Participants: Nicolas Guenot, Lutz Straßburger.

The calculus of structures has mainly be used for "classical" logics that come with a De Morgan duality. The reason is that all normalization procedures developed so far for the calculus of structures rely on this De Morgan duality.

In this work, we give two proof systems for implication-only intuitionistic logic in the calculus of structures. The first is a direct adaptation of the standard sequent calculus to the deep inference setting. It comes with a cut elimination procedure that is similar to the one from the sequent calculus, using a non-local rewriting. The second system is the symmetric completion of the first, as normally given in deep inference for logics with a De Morgan duality: all inference rules have duals, as cut is dual to the identity axiom. For this symmetric system we prove a generalization of cut elimination, that we call symmetric normalization, where all rules dual to standard ones are permuted up in the derivation. The result is a decomposition theorem having cut elimination and interpolation as corollaries. This work has been presented at the CSL-LICS 2014 conference [22].

6.5. Free Theorems for Curry

Participant: Lutz Straßburger.

Free theorems [79] are a means of type-based reasoning and are being successfully applied for typed functional programming languages like Haskell, e.g., for program transformation and generally establishing semantic properties [53], [78]. As a simple example, for every polymorphic function $f :: [\alpha] \to [\alpha]$ from lists to lists, arbitrary types τ_1 and τ_2, and a function $g :: \tau_1 \to \tau_2$, we have $f \circ (\mathsf{map}\ g) = (\mathsf{map}\ g) \circ f$, for the standard function $\mathsf{map} :: (\alpha \to \beta) \to [\alpha] \to [\beta]$ which takes a function and a list and applies that function to every entry of the list. It would be of interest to also have such free theorems available for typed functional-logic languages like Curry.

Previous work [48] has investigated free theorems for such a language, Curry [60], phenomenologically and provides intuition for premises of free theorems as well as counterexamples. Proof of the positive claims has been elusive so far, mainly because Curry's type system fails to reflect the key feature: nondeterminism. This avoidance is convenient for programmers, as they do not have to distinguish between deterministic and nondeterministic values. However, it is a hindrance to formal reasoning: the conditions identified in [48] include a notion of determinism, and hence it is a serious weakness of the type system not to capture this.

In a joint work with colleagues at the University of Bonn, published in [25], we have developed an intermediate language, called SaLT, that allowed us to prove a *Parametricity Theorem* which could be used to derive free theorems for Curry.

This work is the result of the PHC Procope collaboration with the University of Bonn (duration 2012-2013).

6.6. A logical basis for quantum evolution and entanglement

Participant: Lutz Straßburger.

In discrete quantum causal dynamics, quantum systems are viewed as discrete structures, namely directed acyclic graphs. In such a graph, events are considered as vertices and edges depict propagation between events. Evolution is described as happening between a special family of space-like slices, which were referred to as locative slices in [41]. Such slices are not so large as to result in acausal influences, but large enough to capture nonlocal correlations. It was an open problem whether such slices can be captured by a deductive system, such that proof search corresponds to quantum evolution. In a joint work with Blute, Guglielmi, Ivanov, and Panangaden, Straßburger has shown that the logic BV with its mix of commutative and noncommutative connectives, is precisely the right logic for such analysis. More precisely, it was shown that the commutative tensor encodes (possible) entanglement, and the noncommutative *seq* encodes causal precedence. With this interpretation, the locative slices are precisely the derivable strings of formulas. Several new technical results about BV are developed as part of this analysis, which is published in [28]

6.7. On the Pigeonhole and Related Principles in Deep Inference and Monotone Systems

Participant: Anupam Das.

The size of proofs of the propositional pigeonhole principle over various systems is a topic of much interest in the proof complexity literature. In particular, it has received notable attention in recent years from the deep inference community, where its classification over the system KS appears as an open problem in numerous publications. In [21] we construct quasipolynomial-size proofs of the propositional pigeonhole principle in the deep inference system KS, addressing this question by matching the best known upper bound for the more general class of monotone proofs.

We make significant use of monotone formulae computing boolean threshold functions, an idea previously considered in works of Atserias et al. The main construction, monotone proofs witnessing the symmetry of such functions, involves an implementation of merge-sort in the design of proofs in order to tame the structural behavior of atoms, and so the complexity of normalization. Proof transformations from previous work on atomic flows are then employed to yield appropriate KS proofs.

As further results we show that our constructions can be applied to provide quasipolynomial-size KS proofs of the parity principle and the generalized pigeonhole principle. These bounds are inherited for the class of monotone proofs, and we are further able to construct $nO(\log\log n)$-size monotone proofs of the weak pigeonhole principle, thereby also improving the best known bounds for monotone proofs.

6.8. A multi-focused proof system isomorphic to expansion proofs

Participants: Kaustuv Chaudhuri, Stefan Hetzl [Vienna University of Technology, Vienna, Austria], Dale Miller.

The sequent calculus is often criticized for requiring proofs to contain large amounts of low-level syntactic details that can obscure the essence of a given proof. Because each inference rule introduces only a single connective, sequent proofs can separate closely related steps—such as instantiating a block of quantifiers—by irrelevant noise. Moreover, the sequential nature of sequent proofs forces proof steps that are syntactically non-interfering and permutable to nevertheless be written in some arbitrary order. The sequent calculus thus lacks a notion of *canonicity*: proofs that should be considered essentially the same may not have a common syntactic form. To fix this problem, many researchers have proposed replacing the sequent calculus with proof structures that are more parallel or geometric. Proof-nets, matings, and atomic flows are examples of such *revolutionary* formalisms. In [13], we propose, instead, an *evolutionary* approach to recover canonicity within the sequent calculus, which we illustrate for classical first-order logic. The essential element of our approach is the use of a *multi-focused* sequent calculus as the means for abstracting away low-level details from classical cut-free sequent proofs. We show that, among the multi-focused proofs, the *maximally multi-focused* proofs that collect together all possible parallel foci are canonical. Moreover, if we start with a certain focused sequent proof system, such proofs are isomorphic to *expansion proofs*—a well known, minimalistic, and parallel generalization of Herbrand disjunctions—for classical first-order logic. This technique appears to be a systematic way to recover the "essence of proof" from within sequent calculus proofs.

6.9. Equality and fixpoints in the calculus of structures

Participants: Kaustuv Chaudhuri, Nicolas Guenot [IT University of Copenhagen, Denmark].

The standard proof theory for logics with equality and fixpoints suffers from limitations of the sequent calculus, where reasoning is separated from computational tasks such as unification or rewriting. We propose in [20] an extension of the calculus of structures, a deep inference formalism, that supports incremental and contextual reasoning with equality and fixpoints in the setting of linear logic. This system allows deductive and computational steps to mix freely in a continuum which integrates smoothly into the usual versatile rules of multiplicative-additive linear logic in deep inference.

6.10. Automatically deriving schematic theorems for dynamic contexts

Participants: Kaustuv Chaudhuri, Olivier Savary-Bélanger [Princeton University, USA].

Hypothetical judgments go hand-in-hand with higher-order abstract syntax for meta-theoretic reasoning. Such judgments have two kinds of assumptions: those that are statically known from the specification, and the *dynamic assumptions* that result from building derivations out of the specification clauses. These dynamic assumptions often have a simple regular structure of repetitions of *blocks* of related assumptions, with each block generally involving one or several variables and their properties, that are added to the context in a single backchaining step. Reflecting on this regular structure can let us derive a number of structural properties about the elements of the context.

In [26], we present an extension of the Abella theorem prover, which is based on a simply typed intuitionistic reasoning logic supporting (co-)inductive definitions and generic quantification. Dynamic contexts are represented in Abella using lists of formulas for the assumptions and quantifier nesting for the variables, together with an inductively defined *context relation* that specifies their structure. We add a new mechanism for defining particular kinds of regular context relations, called *schemas*, and *tacticals* to derive theorems from these schemas as needed. Importantly, our extension leaves the trusted kernel of Abella unchanged. We show that these tacticals can eliminate many commonly encountered kinds of administrative lemmas that would otherwise have to be proven manually, which is a common source of complaints from Abella users.

6.11. A two-level logic approach for reasoning about typed specification languages

Participants: Kaustuv Chaudhuri, Mary Southern [University of Minnesota, USA].

The *two-level logic approach* (*2LLA*) to reasoning about computational specifications, as implemented by the Abella theorem prover, represents derivations of a *specification language* as an inductive definition in a *reasoning logic*. This approach has traditionally been formulated with the specification and reasoning logics having the *same* type system, and only the formulas being translated. However, requiring identical type systems limits the approach in two important ways: (1) every change in the specification language's type system requires a corresponding change in that of the reasoning logic, and (2) the same reasoning logic cannot be used with two specification languages at once if they have incompatible type systems. In [27], we propose a technique based on *adequate* encodings of the types and judgments of a typed specification language in terms of a simply typed higher-order logic program, which is then used for reasoning about the specification language in the usual *2LLA*. Moreover, a single specification logic implementation can be used as a basis for a number of other specification languages just by varying the encoding. We illustrate our technique with an implementation of the LF dependent type theory as a new specification language for Abella, co-existing with its current simply typed higher-order hereditary Harrop specification logic, without modifying the type system of its reasoning logic.

6.12. Undecidability of multiplicative subexponential logic

Participant: Kaustuv Chaudhuri.

Subexponential logic is a variant of linear logic with a family of exponential connectives—called *subexponentials*—that are indexed and arranged in a pre-order. Each subexponential has or lacks associated structural properties of weakening and contraction. In [18], we show that classical propositional multiplicative linear logic extended with one unrestricted and two incomparable linear subexponentials can encode the halting problem for two register Minsky machines, and is hence undecidable.

6.13. Meta-theoretic results on type isomorphisms in the presence of sums

Participant: Danko Ilik.

Type isomorphisms are a pervasive notion of Theoretical Computer Science. In functional programming, two data types being isomorphic means that we can coerce data and programs back-and-forth between two specifications without loss of information. In Constructive Mathematics, two sets are of the same cardinality exactly when they are isomorphic as types. In the proof theory of intuitionistic logic, two formulas are strongly equivalent precisely when they are isomorphic as types.

However, the theory of simple types made from functions, products, and sums, is well understood only when we do not treat functions and sums at the same time. Fiore, Di Cosmo, and Balat [50], presented a "negative" results: the theory of those type isomorphisms is not finitely axiomatizable. To establish the result, they used the work around the Tarski High School Algebra Problem from Mathematical Logic.

We showed that the picture is not so dark by presenting a positive result: the theory is recursively axiomatizable and decidable. The proofs exploit further the deep theory around Tarski's Problem. This work was presented at the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) in Vienna, Austria [23].

6.14. Towards proof canonicity in presence of disjunction and induction

Participants: Hichem Chihani, Danko Ilik.

The previous work on type isomorphisms showed a way to treat the problem of identity/canonicity of proofs for intuitionistic logic with disjunction, or, equivalently, the problem of the (non-)existence of a canonical eta-long normal form for lambda calculus with if-expressions, which is a long standing open question.

One can see this from the perspective of focusing sequent calculi. The asynchronous phase of proof search is an oriented application of type isomorphisms (by the formulas-as-types correspondence). As we already know that, in the absence of disjunction (sum types), a cut-free focused derivation is eta-long and unique (when the data provided by the synchronous phase is the same), what is necessary in order to handle disjunction is to propagate isomorphisms further than what usual sequent calculus allows. This is related in spirit to deep inference, but more conservative. An implementation of a canonical normalizer and a paper on the topic is under way.

We also intend to use the method to give a proof of focused cut-elimination for the sequent calculi LJF and LKF (at least, for the Sigma-2 fragment) extended with induction. A formal proof in Agda is under development.

6.15. Interpretation of the Sigma-2-classical Axiom of Choice in System T

Participant: Danko Ilik.

Updating previous work, we showed that one can develop a realizability interpretation for the Σ^0_2-fragment of classical Analysis in System T only [36].

This is known to be possible, in principle, by a 1979 result of Schwichtenberg. However, up to day no method that avoids both bar recursion (Spector) and control operators (Krivine) has been known. In fact, we propose to treat control operators as a meta-mathematical technique, rather than to have them in the language of realizers as classical realizability does; we provide a formal proof in Agda that control operators can be completely normalized away from System T while preserving essential equations. [15]

6.16. Axiomatization of constraint systems for first-order reasoning modulo a theory

Participants: Damien Rouhling, Stéphane Graham-Lengrand, Assia Mahboubi, Jean-Marc Notin, Mahfuza Farooque.

This result is part of a work in theorem proving, whose purpose is to provide a theoretical basis for the handling of quantifiers in presence of a theory for which we have specific decision procedures. Inspired by the way first-order unifiers are generated and propagated in automated reasoning techniques such as *tableaux* methods, we sought to generalise these mechanisms to the presence of a theory: We introduced a axiomatic notion of constraint system and a sequent calculus introducing meta-variables and propagating constraints. We then identified the axioms that should be satisfied by the theory's decision procedure, in order for the sequent calculus to be sound and complete. This provides the theoretical basis for the development of Psyche 2.0. This result is submitted for publication.

6.17. Realisability models for cut-elimination in focused systems

Participant: Stéphane Graham-Lengrand.

This result is part of the effort to build meaningful semantics for classical proofs, here based on a polarisation of logical formulae: positive or negative.

Following work by Zeilberger [80], a computational interpretation of cut-elimination in the focused systems LJF and LKF can be given: proofs of positive formulae provide structured data, while proofs of negative formulae consume such data; focusing allows the description of the interaction between the two kinds of proofs as pure pattern-matching.

First, we showed this at a level of abstraction where formulae are no longer made of syntax, yet we also extended the approach so that it could treat quantifiers.

Second, we connected this interpretation to realisability semantics, more precisely orthogonality models, where positive formulae are interpreted as sets of data, and negative formulae are interpreted as their orthogonal sets.

Our construction of orthogonality models for the focused systems LKF and LJF describe the pattern-matching process of cut-elimination in terms of orthogonality. This result has been proved in the Coq proof assistant and forms the second part of [11].

6.18. Refining the FPC framework

Participants: Roberto Blanco, Zakaria Chihani, Quentin Heath, Dale Miller, Fabien Renaud.

We have continued to develop our approach to Foundational Proof Certificates (FPCs). This framework allows defining proof evidence in a general fashion. Proofs in both intuitionistic and classical logics are definable in this framework. We originally have written two different kernels for checking these results but more recently we have found that we can exploit an encoding due to Chaudhuri [43] that enables us to only implement the intuitionistic kernel and then simply encode the classical formulas so that they operator directly on the intuitionistic kernel. This encoding allows for a much more precise and simple means for encoding classical logic into intuitionistic logic than the more familiar double negation translations.

We have also started to develop the second phase of defining proof evidence that was proposed in the ProofCert proposal: the definition of proofs that require fixed points (induction / co-induction). We now have two different kernels being developed on top of the Bedwyr model checker that are checking (and in some cases, proving) theorems involving induction, reachability, and bisimulation.

6.19. Structuring a refinement engine using logic programming

Participants: Dale Miller, Claudio Sacerdoti Coen [University of Bologna], Enrico Tassi [MSR Inria Joint Lab].

The Matita theorem prover is an implementation of the Calculus of Inductive Constructions that is meant to be more accessible (as an implementation) than the Coq system. In an effort to make the Matita kernel more accessible and more flexible, the implementers of that system are experimenting with using a logic programming language similar to λProlog as the control system of the refinement mechanism. In order to use such a logic programming language in this capacity, the notion of flexible goal suspension and *when* declarations are needed. Such a λProlog re-implementation has been written and some experiments in deploying such a system are underway. Formal aspects of λProlog specifications have also been performed using the Abella theorem prover.

7. Partnerships and Cooperations

7.1. European Initiatives

7.1.1. FP7 & H2020 Projects

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.

7.2. International Initiatives

Members of the team have applied for the following three international projects. All three are still pending, the final results are not currently know.

1. A generic ANR proposal for collaboration between several French sites and the University of Bologna.
2. A proposal to ANR and JCJC (Japan).
3. A proposal to the Ministry of Education, Singapore for collaboration with the Nanyang Technological University.

7.3. International Research Visitors

- Chuck Liang (Professor from Hofstra University, NY, USA) visited for three weeks 26 May – 20 June 2014 and for another week starting 15 December.
- Gopalan Nadathur (Professor from the University of Minnesota) visited 2 - 11 July.
- Mary Southern (PhD candidate at the University of Minnesota, USA), May – Aug 2014 Internship supervised by K. Chaudhuri.
- Yuting Wang (PhD candidate at the University of Minnesota, USA), May – Aug 2014

8. Dissemination

8.1. Promoting Scientific Activities

8.1.1. Scientific events organisation

8.1.1.1. General chair, scientific chair

K. Chaudhuri and L. Straßburger co-chaired the 3rd International Workshop on Structures and Deduction (SD), which was associated with the Federated Logic Colloquium (FLoC) and part of the Vienna Summer of Logic (VSL 2014), Vienna, Austria.

D. Miller was a PC chair for the Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL14) and the 29th ACM/IEEE Symposium on Logic in Computer Science (LICS14), 14-18 July, Vienna.

8.1.1.2. Member of the organizing committee

K. Chaudhuri was Publicity co-chair for CSL-LICS 2014.

8.1.2. Scientific events selection

8.1.2.1. Member of the conference program committee

K. Chaudhuri was on the Program Committees of:

30th International Conference on Logic Programming (ICLP)

4th ACM-SIGPLAN Conference on Certified Programs and Proofs (CPP)

9th International Workshop on Logical Frameworks and Meta-languages: Theory and Practice (LFMTP)

3rd International Workshop on Linearity

D. Ilik was on the Program Committee of the 16th International Symposium on Principles and Practice of Declarative Programming (PPDP 2014), Canterbury, United Kingdom.

D. Miller was on the Program Committee of WoLLIC 2014: Workshop on Logic, Language, Information and Computation, Valparaiso Chile, 1-4 September.

L. Straßburger was the Program Committee of the 18th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS).

8.1.2.2. Reviewer

S. Graham-Lengrand was a reviewer for the following conferences:

International Joint Conference on Automated Reasoning (IJCAR 2014)

Conferences on Intelligent Computer Mathematics (CICM 2014).

L. Straßburger was reviewer for the following conferences:

7th International Joint Conference on Automated Reasoning (IJCAR 2014)

41st International Colloquium on Automata, Languages and Programming (ICALP 2014)

Joint meeting of the 23rd EACSL Annual Conference on Computer Science Logic and the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (CSL-LICS 2014)

F. Lamarche was a reviewer for the following conferences:

Joint International Conferences, Rewriting Techniques and Applications, Typed Lambda Calculus and Applications (RTA - TLCA 2014)

18th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS).

D. Ilik was a reviewer for the following conferences:

16th International Symposium on Principles and Practice of Declarative Programming (PPDP 2014), Canterbury, United Kingdom

18th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2015)

20th Conference "Types for Proofs and Programs" (TYPES 2014)

24th European Symposium on Programming (ESOP 2015)

Joint International Conferences, Rewriting Techniques and Applications, Typed Lambda Calculus and Applications (RTA - TLCA 2014)

8.1.3. Journal

8.1.3.1. Editor-in-chief

D. Miller was Editor-in-Chief of the ACM Transactions on Computational Logic.

8.1.3.2. Member of the editorial board

S. Graham-Lengrand was guest editor for the special issue on computational logic in honour of Roy Dyckhoff, in the *Journal of Logic and Computation* (2014) [29]

D. Miller is on the editorial board of the following journals: *ACM Transactions on Computational Logic*, *Journal of Automated Reasoning* (Springer), *Theory and Practice of Logic Programming* (Cambridge University Press), and *Journal of Applied Logic* (Elsevier).

8.1.3.3. reviewer

S. Graham-Lengrand was a reviewer for the following journals:

Logic Journal of the IGPL (OUP)

Information and Computation (Elsevier)

Acta Informatica (Springer)

Theoretical Computer Science (Elsevier).

L. Straßburger was reviewer for the following journals:

Archive for Mathematical Logic (Springer)

ACM Transactions on Computational Logic (ACM-ToCL)

F. Lamarche was reviewer for the following journal:

Annals of Pure and Applied Logic (Elsevier).

D. Ilik was reviewer for the following journals:

Logic Journal of the IGPL (OUP)

Mathematical Reviews (MathSciNet)

Theoretical Computer Science (Elsevier)

Zentralblatt MATH

D. Miller was reviewer for the Studia Logica journal.

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

Licence: K. Chaudhuri, "*INF 321 : Les principes des langages de programmation*", 40 hours eq. TD, L3, Ecole polytechnique, France

Licence: S. Graham-Lengrand, "*INF431: Algorithmique et programmation*", 50 hours eq. TD, L3, Ecole Polytechnique, France.

Licence: S. Graham-Lengrand, "*INF412: Les bases de la programmation et de l'algorithmique*", 32 hours eq. TD, L3, Ecole Polytechnique, France.

Master: S. Graham-Lengrand, "*INF551: Computer-aided reasoning*", 36 hours eq. TD, M1, Ecole Polytechnique, France.

Master: S. Graham-Lengrand, "*Curry-howard correspondence for classical logic*", 12 hours eq. TD, M2, Master Parisien de Recherche en Informatique, France.

Master: D. Miller, "*MPRI 2-1: Logique linéaire et paradigmes logiques du calcul*", 12 hours, M2, Master Parisien de Recherche en Informatique, France.

8.2.2. Supervision

PhD: Hernán Vanzetto, "*Automatisation des preuves et synthèse des types pour la théorie des ensembles dans le contexte de TLA$^+$*", Université de Lorraine, 4 Dec 2014; supervised by Stephan Merz (Inria Lorraine) and K. Chaudhuri

PhD in progress: Sonia Marin, 1 Nov 2014, supervised by L. Straßburger and D. Miller

PhD in progress: Roberto Blanco, Zakaria Chihani, and Quentin Heath, supervised by D. Miller

8.2.3. Juries

L. Straßburger was external reviewer and jury member for the PhD of Roman Krenický, University of Manchester, UK. Defence date: April 28, 2014.

S. Graham-Lengrand was a jury member for the PhD of Luis Pino Duque, École Polytechnique, France. Defence date: September 25, 2014.

D. Miller was a reporter for the following two PhD theses: Taus Brock-Nannestad (IT University of Copenhagen, 12 September 2014) and Thanos Tsouanas (ENS Lyon, 2 July 2014).

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

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

[4] 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

[5] 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

[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] S. GRAHAM-LENGRAND. *Polarities & Focussing: a journey from Realisability to Automated Reasoning*, Paris-Sud XI, December 2014, Habilitation à diriger des recherches, https://tel.archives-ouvertes.fr/tel-01094980

Articles in International Peer-Reviewed Journals

[12] A. ASCHERIO, K. L. MUNGER, R. WHITE, K. KÖCHERT, K. C. SIMON, C. H. POLMAN, M. S. FREEDMAN, H.-P. HARTUNG, D. H. MILLER, X. MONTALBÁN, G. EDAN, F. BARKHOF, D. PLEIMES, E.-W. RADÜ, R. SANDBRINK, L. KAPPOS, C. POHL. *Vitamin D as an early predictor of multiple sclerosis activity and progression*, in "JAMA Neurology", February 2014, pp. 306-14, http://www.hal.inserm.fr/inserm-01103094

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

[14] W. HEIJLTJES, L. STRASSBURGER. *Proof nets and semi-star-autonomous categories*, in "Mathematical Structures in Computer Science", November 2014, pp. 1-40 [*DOI :* 10.1017/S0960129514000395], https://hal.inria.fr/hal-01092253

[15] D. ILIK, K. NAKATA. *A Direct Version of Veldman's Proof of Open Induction on Cantor Space via Delimited Control Operators*, in "Leibniz International Proceedings in Informatics (LIPIcs)", December 2014, pp. 288-201 [*DOI :* 10.4230/LIPICS.TYPES.2013.188], https://hal.inria.fr/hal-01092427

[16] F. LAMARCHE. *Modeling Martin Löf Type Theory in Categories*, in "Journal of Applied Logic", 2014, vol. 12, n^{o} 1, pp. 28-44 [*DOI :* 10.1016/J.JAL.2013.08.003], https://hal.inria.fr/hal-00706562

[17] G. J. A. NAGTEGAAL, C. POHL, M. P. WATTJES, H. E. HULST, M. S. FREEDMAN, H.-P. HARTUNG, D. MILLER, X. MONTALBÁN, L. KAPPOS, G. EDAN, D. PLEIMES, K. BECKMAN, B. STEMPER, C. H. POLMAN, R. SANDBRINK, F. BARKHOF. *Interferon beta-1b reduces black holes in a randomised trial of clinically isolated syndrome*, in "Mult Scler", January 2014, vol. 20, n^{o} 2, pp. 234-42, Trial Registration Number: NCT00544037, http://www.hal.inserm.fr/inserm-01103080

International Conferences with Proceedings

[18] K. CHAUDHURI. *Undecidability of Multiplicative Subexponential Logic*, in "3rd International Workshop on Linearity", Vienna, Austria, S. ALVES, I. CERVESATO (editors), July 2014, https://hal.inria.fr/hal-00998753

[19] K. CHAUDHURI, M. CIMINI, D. MILLER. *A Lightweight Formalization of the Metatheory of Bisimulation-Up-To*, in "4th ACM-SIGPLAN Conference on Certified Programs and Proofs (CPP)", Mumbai, India, X. LEROY, A. TIU (editors), ACM Proceedings, January 2015 [*DOI :* 10.1145/2676724.2693170], https://hal.inria.fr/hal-01091524

[20] K. CHAUDHURI, N. GUENOT. *Equality and fixpoints in the calculus of structures*, in "Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)", Vienna, Austria, T. A. HENZINGER, D. MILLER (editors), ACM-SIGPLAN, July 2014, pp. 1 - 10 [*DOI :* 10.1145/2603088.2603140], https://hal.inria.fr/hal-01091570

[21] A. DAS. *On the pigeonhole and related principles in deep inference and monotone systems*, in "CSL-LICS '14 Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)", Vienna, Austria, July 2014, pp. 1 - 10 [*DOI :* 10.1145/2603088.2603164], https://hal.inria.fr/hal-01096154

[22] N. GUENOT, L. STRASSBURGER. *Symmetric Normalisation for Intuitionistic Logic*, in "CSL-LICS 2014", Vienna, Austria, July 2014 [*DOI :* 10.1145/2603088.2603160], https://hal.inria.fr/hal-01092105

[23] D. ILIK. *Axioms and Decidability for Type Isomorphism in the Presence of Sums*, in "CSL-LICS '14 Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)", Vienna, Austria, T. HENZINGER, D. MILLER (editors), ACM, July 2014 [*DOI :* 10.1145/2603088.2603115], https://hal.inria.fr/hal-00991147

[24] S. MARIN, L. STRASSBURGER. *Label-free Modular Systems for Classical and Intuitionistic Modal Logics*, in "Advances in Modal Logic 10", Groningen, Netherlands, August 2014, https://hal.inria.fr/hal-01092148

[25] S. MEHNER, D. SEIDEL, L. STRASSBURGER, J. VOIGTLÄNDER. *Parametricity and Proving Free Theorems for Functional-Logic Languages*, in "Principles and Practice of Declarative Programming - PPDP",

Canterbury, United Kingdom, September 2014 [*DOI :* 10.1145/2643135.2643147], https://hal.inria.fr/hal-01092357

[26] O. SAVARY BÉLANGER, K. CHAUDHURI. *Automatically Deriving Schematic Theorems for Dynamic Contexts*, in "Logical Frameworks and Meta-Languages: Theory and Practice", Vienna, Austria, ACM-SIGPLAN, July 2014 [*DOI :* 10.1145/2631172.2631181], https://hal.inria.fr/hal-01091555

[27] M. SOUTHERN, K. CHAUDHURI. *A Two-Level Logic Approach to Reasoning about Typed Specification Languages*, in "34th International Conference on Foundation of Software Technology and Theoretical Computer Science (FSTTCS 2014)", New Delhi, India, V. RAMAN, S. P. SURESH (editors), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, December 2014, vol. 29 [*DOI :* 10.4230/LIPICS.FSTTCS.2014.557], https://hal.inria.fr/hal-01091544

Scientific Books (or Scientific Book chapters)

[28] R. BLUTE, A. GUGLIELMI, I. T. IVANOV, P. PANANGADEN, L. STRASSBURGER. *A Logical Basis for Quantum Evolution and Entanglement*, in "Categories and Types in Logic, Language, and Physics - Essays Dedicated to Jim Lambek on the Occasion of His 90th Birthday", Springer, 2014, vol. 8222, pp. 90 - 107 [*DOI :* 10.1007/978-3-642-54789-8_6], https://hal.inria.fr/hal-01092279

[29] D. GALMICHE, S. GRAHAM-LENGRAND. *Special issue on computational logic in honour of Roy Dyckhoff. Journal of Logic and Computation*, Oxford University Press, June 2014, https://hal.archives-ouvertes.fr/hal-01094032

[30] T. HENZINGER, D. MILLER. *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, ACM, July 2014, pp. 1-764, https://hal.inria.fr/hal-01087515

Scientific Popularization

[31] D. BAELDE, K. CHAUDHURI, A. GACEK, D. MILLER, G. NADATHUR, A. TIU, Y. WANG. *Abella: A System for Reasoning about Relational Specifications*, in "Journal of Formalized Reasoning", 2014, vol. 7, n^{o} 2, pp. 1-89 [*DOI :* 10.6092/ISSN.1972-5787/4650], https://hal.inria.fr/hal-01102709

Other Publications

[32] R. ARISAKA. *Gradual Classical Logic for Attributed Objects*, 2014, https://hal.inria.fr/hal-00969271

[33] R. ARISAKA. *Structural Interactions and Absorption of Structural Rules in BI Sequent Calculus*, April 2014, https://hal.inria.fr/hal-00982331

[34] R. ARISAKA, A. DAS, L. STRASSBURGER. *On Nested Sequents for Constructive Modal Logics*, 2014 [*DOI :* 10.2168/LMCS-???], https://hal.inria.fr/hal-01093143

[35] R. DAMIEN, M. FAROOQUE, S. GRAHAM-LENGRAND, J.-M. NOTIN, A. MAHBOUBI. *Axiomatisation of constraint systems to specify a tableaux calculus modulo theories*, December 2014, Preprint, https://hal.inria.fr/hal-01107944

[36] D. ILIK. *An interpretation of the Sigma-2 fragment of classical Analysis in System T*, December 2014, https://hal.inria.fr/hal-01092411

References in notes

[37] J. A. ROBINSON, A. VORONKOV (editors). *Handbook of Automated Reasoning*, Elsevier and MIT press, 2001

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

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

[40] 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° 6173, pp. 278–292 [*DOI :* 10.1007/978-3-642-14203-1], http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/ijcar10.pdf

[41] R. BLUTE, I. IVANOV, P. PANANGADEN. *Discrete quantum causal dynamics*, in "International Journal of Theoretical Phsysics", 2003, vol. 42, pp. 2025–2041

[42] 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

[43] K. CHAUDHURI. *Classical and Intuitionistic Subexponential Logics are Equally Expressive*, in "19th Annual EACSL Conference on Computer Science Logic (CSL 2010)", Brno, Czech Republic, A. DAWAR, H. VEITH (editors), LNCS, Springer, August 2010, vol. 6247, pp. 185–199 [*DOI :* 10.1007/978-3-642-15205-4_17]

[44] 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 [*DOI :* 10.4230/LIPICS.CSL.2011.159], http://drops.dagstuhl.de/opus/volltexte/2011/3229/pdf/16.pdf

[45] K. CHAUDHURI, S. HETZL, D. MILLER. *A Multi-Focused Proof System Isomorphic to Expansion Proofs*, in "Journal of Logic and Computation", June 2014 [*DOI :* 10.1093/LOGCOM/EXU030], http://hal.inria.fr/hal-00937056

[46] Z. CHIHANI, D. MILLER, F. RENAUD. *Checking Foundational Proof Certificates for First-Order Logic (extended abstract)*, in "Third International Workshop on Proof Exchange for Theorem Proving (PxTP 2013)", J. C. BLANCHETTE, J. URBAN (editors), EPiC Series, EasyChair, 2013, vol. 14, pp. 58–66

[47] Z. CHIHANI, D. MILLER, F. RENAUD. *Foundational proof certificates in first-order logic*, in "CADE 24: Conference on Automated Deduction 2013", M. P. BONACINA (editor), Lecture Notes in Artificial Intelligence, 2013, n° 7898, pp. 162–177

[48] J. CHRISTIANSEN, D. SEIDEL, J. VOIGTLÄNDER. *Free Theorems for Functional Logic Programs*, in "PLPV, Proceedings", ACM, 2010, pp. 39–48

[49] C. DROSS, S. CONCHON, J. KANIG, A. PASKEVICH. *Reasoning with Triggers*, in "10th Intern. Worksh. on Satisfiability Modulo Theories, SMT 2012", P. FONTAINE, A. GOEL (editors), EPiC Series, EasyChair, June 2012, vol. 20, pp. 22–31, http://www.easychair.org/publications/?page=2135488790

[50] M. FIORE, R. D. COSMO, V. BALAT. *Remarks on isomorphisms in typed lambda calculi with empty and sum types*, in "Annals of Pure and Applied Logic", 2006, vol. 141, pp. 35–50

[51] P. FONTAINE, J.-Y. MARION, S. MERZ, L. P. NIETO, A. TIU. *Expressiveness + Automation + Soundness: Towards Combining SMT Solvers and Interactive Proof Assistants*, in "TACAS: Tools and Algorithms for the Construction and Analysis of Systems, 12th International Conference", H. HERMANNS, J. PALSBERG (editors), LNCS, Springer, 2006, vol. 3920, pp. 167–181 [*DOI :* 10.1007/11691372_11]

[52] 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

[53] A. GILL, J. LAUNCHBURY, S. PEYTON JONES. *A short cut to deforestation*, in "FPCA, Proceedings", ACM, 1993, pp. 223–232

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

[55] S. GRAHAM-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

[56] 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

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

[58] 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

[59] 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

[60] M. HANUS. *Functional Logic Programming: From Theory to Curry*, in "Programming Logics — Essays in Memory of Harald Ganzinger", LNCS, Springer, 2013, vol. 7797, pp. 123–168

[61] R. HARPER, F. HONSELL, G. PLOTKIN. *A Framework for Defining Logics*, in "2nd Symp. on Logic in Computer Science", Ithaca, NY, June 1987, pp. 194–204

[62] 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 [*DOI :* 10.1016/J.TCS.2009.07.041], http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/tcs09.pdf

[63] C. LIANG, D. MILLER. *A Focused Approach to Combining Logics*, in "Annals of Pure and Applied Logic", 2011, vol. 162, n^{o} 9, pp. 679–697 [*DOI :* 10.1016/J.APAL.2011.01.012]

[64] 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

[65] 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

[66] 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

[67] D. MILLER. *A Multiple-Conclusion Meta-Logic*, in "9th Symp. on Logic in Computer Science", Paris, S. ABRAMSKY (editor), IEEE Computer Society Press, July 1994, pp. 272–281

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

[69] D. MILLER, G. NADATHUR. *Programming with Higher-Order Logic*, Cambridge University Press, June 2012 [*DOI :* 10.1017/CBO9781139021326]

[70] 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

[71] 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

[72] 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

[73] 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

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

[75] L. STRASSBURGER. *Cut Elimination in Nested Sequents for Intuitionistic Modal Logics*, in "FoSSaCS'13", F. PFENNING (editor), LNCS, Springer, 2013, vol. 7794, pp. 209–224

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

[77] 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

[78] J. VOIGTLÄNDER. *Bidirectionalization for Free!*, in "POPL, Proceedings", ACM, 2009, pp. 165–176

[79] P. WADLER. *Theorems for free!*, in "FPCA, Proceedings", ACM, 1989, pp. 347–359

[80] N. ZEILBERGER. *The Logical Basis of Evaluation Order and Pattern-Matching*, Carnegie Mellon University, April 2009