INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team TypiCal

Types, Logic, and Computation

Saclay - Île-de-France

Theme : Programs, Verification and Proofs

Table of contents

The TypiCal project is a common project gathering researchers from INRIA Saclay – Île-de-France sud, École Polytechnique, and CNRS at LIX. The team leader is Benjamin Werner.

1. Team

Research Scientist

Benjamin Werner [DR INRIA, Team Leader]
Bruno Barras [CR INRIA]
Germain Faure [CR INRIA]
Assia Mahboubi [CR INRIA]

Faculty Member

Gilles Dowek [Professor, École Polytechnique, HdR]

Technical Staff

Jean-Marc Notin [IR CNRS]

PhD Student

Lisa Allali [Région Ile-de-France, until September 2009]
Bruno Bernardo [DGA until October 2009, ATER Paris 7]
Eric Biagoli [Allocation INRIA-Saclay IdF, since September 2009]
Mathieu Boespflug [AMN]
Cyril Cohen [Allocataire Ministère, since September 2009]
Denis Cousineau [MENRT until September 2009]
Chantal Keller [ENS Lyon, since September 2009]
Arnaud Spiwack [ENS Cachan]

Post-Doctoral Fellow

Benoît Valiron [until September 2009]

Administrative Assistant

Lydie Fontaine [TR INRIA]

Other

Eric Biagoli [Master Internship]
Cyril Cohen [Master Internship]
Chantal Keller [Master Internship]

2. Overall Objectives

2.1. Presentation

Mathematics is among the many human activities that have been transformed by the invention of the computer and its broad diffusion in the second half of the 20^{th} century. Mathematicians could, from then on, use a tool allowing to carry out operations that were too long or too tedious to be executed by hand. Like the use of the telescope in astronomy, the use of the computer opened many new prospects in mathematics. One of these prospects is the use of *proof assistants*, *i.e.* computer programs which perform some operations on mathematical proofs. The goal of the research developed in the TypiCal project-team is to develop such *proof assistants*. The main effort of the project-team is to contribute to the development proof assistants in general and of the Coq system in particular, which has an important community of users in industry and in academia. However, we believe that the development of a proof assistant cannot be accomplished without a joint reflection about the structure of mathematical proofs and about the use of proof assistants in various applicative domains. We also believe that proof assistants should take benefit of the use of automated deduction tools. Thus, the questions addressed in the team range from questions related to the Coq system, such as "What will be the features of the next version of Coq?", to more theoretical questions of logic, such as "What is a proof?" and more applied ones, such as "How can I delegate part of the proof search to automated tools?" or "How can we use a proof assistant to check whether a protocol is free of deadlocks?".

2.2. Highlights of the year

Denis Cousineau defended his PhD [1] in December 2009 . It opens new possibilities for the study of proof normalization by giving the first model-based definition of consistency.

Three new PhD students integrated the team. Cyril Cohen is working on algebraic numbers in Coq. Chantal Keller is working the cooperation of SMT solvers with Coq. Eric Biagoli is working on global optimisation.

The version 8.2 of the Coq system has been released. Also, the ssreflect proof tactics package has been released by the INRIA MSR joint centre with an important contribution of TypiCal researchers [17].

The first version of Dedukti, a universal proof checker, has been released.

3. Scientific Foundations

3.1. Proof assistants

The first operation that a proof assistant can perform on a proof is to check its correctness. This participates in the quest for a new step in mathematical rigor: the point where nothing is understated, and where the reader can therefore be replaced by a program. This quest for rigor is specially important for the large proofs, either hand written or computer aided, that mathematicians have built since the middle of the 20^{th} century. For instance, without using a proof assistant, it is quite difficult to establish the correctness of a proof using symbolic computations on polynomials formed with hundreds of monomials, or a case analysis requiring the inspection of several hundreds of cases, or establishing that a complex object such as a long program or a complex digital circuit has some property. This quest for correctness is especially important in application domains where a malfunction may jeopardize human life, health or environment, such as transportations or computer aided surgery.

Besides this correctness check, proof assistants can help the users to build proofs interactively. The “tactic language” allowing the user to control the system in this proof construction process has always been the object of intensive studies. The ML language, for instance, was originally the tactic language of the LCF proof assistant. More recent questions about this language are focused on the formal expression of its operational semantic, in particular the handling of exceptions.

Proof assistants may also prove some easy but big size lemmas automatically. The automatisation of proof assistants can be increased by the development of decision procedures. Either they can be developed inside the proof assistant or we can use external tools producing certificates later used inside the proof assistant to reconstruct the proof.

Proof assistants may also transform mathematical proofs into other formal objects such as programs.

A more recent kind of applications is the construction of large libraries of mathematical results on the net.

3.2. Formalization of mathematics

A proof assistant implements a particular formalism allowing to express mathematics. A traditional formalism allowing to express mathematics is set theory, built on top of first-order predicate logic. Unfortunately, this formalism does not address exactly the needs of a proof assistant. Set theory has been elaborated at the beginning of the 20^{th} century to study mathematically the properties of mathematical reasoning. For this purpose, being able to formalize mathematics “in principle” was enough. Nowadays, the problem is not to formalize mathematics “in principle” but to formalize them “in facts”. Thus, the design of proof assistants has led to ask new questions in logic and, in particular, in proof theory.

Several variants or alternative to set theory have been designed to express mathematics in practice. The system Coq is based on a formalism called *The Calculus of Inductive Constructions*.

An important feature for such a formalism is the language allowing to express mathematical objects such as functions and sets. It is not desirable to use a formalization of mathematics that has only existence axioms, or even one having the combinator's language obtained by skolemizing these axioms in predicate logic. It is important to have a rich and compact language, in particular a language with binders such as the λ-calculus.

Another important feature is the ability to integrate deduction and computation. It is not possible, when we use a proof assistant to consider that the proposition $2 + 2 = 4$ requires a proof, even a proof simple enough to be found by a automated theorem proving system. Several formalisms such as Martin-Löf's type theory, Boyer-Moore logic, the Calculus of Constructions and the Calculus of Inductive Constructions, include such a possibility to compute inside a proof. Thus, these formalisms designed to express mathematics contain a programming language as a sub-language.

More recently the research in this area has taken several different directions: first the study of *deduction modulo* that is the simplest extension of predicate logic allowing to mix deduction and computation. Deduction modulo has applications both in automated theorem proving and in proof theory, where it paves the way to a unified theory of cut elimination. Finally, the need to improve the efficiency of computations in the system Coq, has led to the use of compilation techniques issued from the theory of programming language. This has brought logical languages and programming languages closer, allowing for instance to use the language of Coq as a general purpose programming language. This perspective of unifying proof and programming languages is a real challenge for future proof assistants.

Another property of the Calculus of Inductive Constructions that is important for its use as the language of a proof assistant is the possibility to write both constructive and classical proofs. When a proof of existence is constructive, the user can request the computation of a witness, but, of course, not when it is classical.

By insisting on this idea that *constructive proofs* must be distinguished from classical proofs, the project-team TypiCal participates to rise of a new form a constructivism, not trying to restrict mathematics to constructive mathematics, but trying to identify the part of mathematics that can be done constructively and the part that cannot.

A last property of the Calculus of Inductive Constructions is that proofs are objects of the formalism, exactly as numbers, functions and sets are. This property, based on the celebrated Curry-De Bruijn-Howard correspondence, allows to reduce the safety critical base of the Coq system to a quite small kernel.

4. Application Domains

4.1. Application Domains

The first application is to pure mathematics. The use of proof assistants for proving genuine mathematical theorems has been considered as utopic for long. But several recent developments have changed the situation. First of all, the development of libraries of both constructive and classical analysis has led the possibility to use Coq, not only in remote areas of discrete mathematics, but also to prove mainstream mathematical theorem as taught in an undergrad textbook for instance. This direction culminated with the proof in Coq of the Fundamental Theorem of Algebra, a few years ago, by a group of researchers in Nijmegen. More recent work include a proof of the Four color theorem in Coq, proofs of lemma's on polynomials used in the proof of Hale's Sphere packing theorem (Kepler's conjecture), proofs in algebraic geometry by a group of mathematicians in Nice. The Mathematical Components group of the INRIA - MSR Joint Centre is working on the formalisation of the Feit Thompson theorem (1962) for groups of odd order, which is a milestone in the classification of finite groups.

Another direction is the proof of algorithms. In proofs of algorithms (as opposed to proofs of programs) a property is proved on an algorithms formalized in the language of Coq. An example is the recent proof of algorithms used in floating point arithmetic or the older proof carried out by the company *Trusted Logic* of the correctness that has reached, for the first time, the EAL7 level in common criteria.

The most applied use of Coq is the proof of programs where an actual program written in the syntax of a general purpose programming language (such as Caml, Java or C). The system Coq is used by the ProVal project-team, that has strong historical connections to TypiCal, as a back-end of their systems Why, Krakatoa and Caduceus.

5. Software

5.1. Coq

The TypiCal team participates to the developments of the *Coq* system. The *Coq* system is a processor of mathematical proofs allowing an interactive development of specifications and proofs.

At the architectural level, the main feature is the isolation of the critical code preforming the proof checking in a kernel small enough to reach higher levels of reliability of the whole system (with the current goal of achieving the self-validation), and the production of an abstract interface of that kernel granting that theories can only be built using the features of the kernel. A standalone checker of compiled libraries can be used to validate libraries with an even higher level of confidence.

Coq is used in hundreds of sites. We have demanding users in industry (France Télécom R & D, Dassault-Aviation, Trusted Logic, Gemplus, Schlumberger-Sema, ...) in the academic world in Europe (Scotland, Netherlands, Spain, Italy, Portugal, ...) and in France (Bordeaux, Lyon, Marseille, Nancy, Nantes, Nice, Paris, Strasbourg, ...).

The *Coq* system is available from URL http://coq.inria.fr/. Written in Objective Caml and Camlp4, it is ported to most Unix architectures, but also to Windows and MacOS.

5.2. Dedukti

A universal proof checker called DEDUKTI has been realeased. This is an implementation of a type checker for proofs written in the $\lambda\Pi$-modulo calculus.

To make the translator useful in practice, we are currently writting different translator from already existing proof assistants to DEDUKTI. Our effort are now concentrated on the translation from Coq. This raised difficult theoretical questions that are under investigation.

DEDUKTI is available from ULR http://www.lix.polytechnique.fr/dedukti. It is written in HASKELL and is short enough to be trustable. It is a product of a long run collaboration with other INRIA teams. This collaboration was formalized by an INRIA ARC (special funding for collaborated INRIA teams).

6. New Results

6.1. Development of theories and tactics

6.1.1. First-order linear arithmetics, linear and convex optimisation

Participants: Assia Mahboubi, Salil Joshi, Pierre-Yves Strub, Eric Biaggioli, Benjamin Werner.

Assia Mahboubi has supervised Salil Joshi's internship on the development of a reflexive tactic for first order linear integer arithmetic for the Coq system. This work has lead to a full implementation and formal proof of correctness of the so-called Cooper algorithm. Optimization of this code is work in progress.

Assia Mahboubi has worked with Pierre-Yves Strub (INRIA Rocquancourt, Tsinghua University) around the use of linear programming in Coq. This work has two aspects. The first one is to use an OCaml implementation of the simplex algorithm to provide certificates for unsatisfiability of linear arithmetic problems without quantifiers. Using a simplex with a trivial cost function is a usual way to decide this theory, in particular for SMT tools. But this also provide small and convenient certificated that can be checked formally. This has been used to implement an efficient proof- producing decision procedure for the Coq system, enhancing previous work done by Frederic Besson. This tactic could for instance be used to verify SMT output traces. This work is done in the context of the DECERT ANR project. The second aspect of this work is a formalization of the theory behind the simplex algorithm. We ensure the correctness of a straightforward implementation of the simplex algorithm in Coq. This formalization deals with convexity (with a proof of the weak Krein-Milman theorem), and linear optimization. It has been carried in the Coq system, on top of the ssreflect extension developed in the Mathematical Component team.

Under the supervision of Benjamin Werner and Stéphane Gaubert (Maxplus), Eric Biaggioli is working on techniques coming from convex optimization, in order to formally prove in Coq inequalities over real expressions, especially polynomials.

6.1.2. Binders

Participants: Jean-Marc Notin, Benjamin Werner, Gilles Dowek, Murdoch James Gabbay, Dominic Mulligan.

Benjamin Werner and Jean-Marc Notin are working on a formalization of Normalization by Evaluation for simply typed λ-Calculus in Coq, with the aim of proposing a convenient way to handle expressions with binders in formal proofs.

Gilles Dowek, Jamie Gabbay, and Dominic Mulligan, have proposed a new syntax for terms with binders and shown how unification problems on these terms could be translated to higher-order pattern unification problems [9].

6.1.3. Hierarchy of algebraic structures

Participant: Assia Mahboubi.

Assia Mahboubi participates to the Mathematical Component project at the INRIA Microsoft Research Joint Centre. The aim of this project is to investigate formal proofs software engineering methods, which means finding good patterns leading to modular, reusable formal libraries. This project validates this research by building a formalization of the proof of the Feit-Thompson theorem (also called Odd Order theorem), which is a corner-stone of the classification of finite simple groups. The extension of the Coq system, as well as the libraries distributed with this extension are altogether called the ssreflect extension.

In this context, Assia Mahboubi has worked with Georges Gonthier (Microsoft Research) on the development of a hierarchy of algebraic structures in the Coq system, using the ssreflect extension. In particular, she has developed the structure and theory of decidable rings and rings admitting quantifier elimination. This work has led to a publication at the TPHOLs 2009 conference [10].

6.1.4. Formalisation of finite group theory

Participant: Assia Mahboubi.

Assia Mahboubi has also pursued her work on the formalization of finite group theory. She has generalized her previous proof of the Jordan Hoder theorem, in particular in order to introduce the notion of chief factors. This work is the basis needed for the Hall theorem of the first chapter of the Local Analysis for the Odd Order Theorem. This volume describes the local analysis part of the Feit Thompson theorem proof. This Hall theorem and its corollaries is the only part missing for this first chapter to be completely formalized.

6.1.5. Quotient types

Participants: Assia Mahboubi, Cyril Cohen.

Assia Mahboubi has supervised the master (MPRI) internship of Cyril Cohen. Cyril Cohen has worked on the formalization of quotient types in the Calculus of Inductive Constructions.

6.1.6. Higher-order matching

Participant: Germain Faure.

Germain Faure studied higher-order matching in an untyped setting while the standard approach uses typed setting. He showed that this is particularly interesting because (1) an easy and efficient algorithm can be build (2) second-order matching is subsumed by the problems we deal with. He also showed that these results can be applied with success in the context of higher-order rewriting. This work leads to a research report [15].

6.1.7. Formal semantics of the Calculus of Constructions and extensions

Participant: Bruno Barras.

Bruno Barras has formalized a simple set theoretical (proof-irrelevant) model of the Calculus of Constructions with an infinite hierarchy of universes [4]. For this purpose, several set theories were modelized in Coq: hereditarily finite sets and ZF. In those theories a piece of ordinal theory was developed.

The logical consistency of the Calculus of Constructions can be derived from the soundness result for this model. A simple modification of the model leads to a "strong normalization model" of the Calculus of Constructions with universes.

He also made a first step towards formalizing the semantics of the Calculus of Inductive Constructions by studying the extension of those formalisms with natural numbers and ordinals, sticking very closely to the inductive types used to model them. The main novelty is that the elimination principles are split into a case-analysis principle and a general fixpoint operator. The totality of the latter being warranted by size annotations on inductive types.

6.2. Developments of systems

6.2.1. Translation of HOL-light proofs into Coq

Participants: Chantal Keller, Benjamin Werner.

Chantal Keller and Benjamin Werner have designed a new way to translate HOL-proofs and import them into Coq. This translation has been implemented by Chantal Keller and is operational. It is based on ideas coming from a previous implementation of normalization-by-evaluation in Coq, by Benjamin Werner and François Garillot. The results of the implementation are encouraging; a pleasing particularity of this translation is that the theorems are translated into Coq in an intelligible and reusable form.

6.2.2. Interfacing Coq with SMT solvers

Participants: Germain Faure, Chantal Keller, Benjamin Werner, Assia Mahboubi.

The starting point of this work is to note that SMT solvers, deciding the Satisfiability Modulo Theories, are in constant evolution to take into account new decision procedures as well as theories. These systems are rather complex and it is now clearly established that they all contain bugs. The standard approach is to ask the SMT solver to append to the decision result a certificate that can be checked by another tool.

In this context, we are using formal systems like Coq to check the certificate. The first experiments we have done in close collaboration with the Marelle team (INRIA Sophia Antipolis) are clearly promising.

6.2.3. A new engine to interpret Coq tactics

Participant: Arnaud Spiwack.

Arnaud Spiwack implemented a new engine to interpret Coq tactics. It gives new functionalities (structured proofs based on bullets, true refinement, non-determinism in tactics...) and the overall development will be part of the 2010 release of Coq.

6.2.4. Implementation of the lambda Pi-modulo calculus

Participant: Mathieu Boespflug.

Mathieu Boespflug wrote a full implementation of a type checker for proofs written in the $\lambda\Pi$-modulo calculus, called DEDUKTI. Version 1.0 of DEDUKTI was released in September.

6.2.5. Translator of Coq proofs into lambda Pi-modulo proofs

Participants: Mathieu Boespflug, Denis Cousineau.

Denis Cousineau has written a prototype called COQINE for a translator from Coq proofs to proofs in the $\lambda\Pi$-calculus modulo. This translator was further developed by Guillaume Burel and Mathieu Boespflug. COQINE has not yet seen a formal released and is under active development. Dependant pattern-matching has yet to be supported, as well as other advanced Coq features such as modules.

6.2.6. Prototyping programming languages

Participants: Gilles Dowek, César Muñoz, Camilo Rocha.

Gilles Dowek, César Muñoz and Camilo Rocha have proposed an environment to prototype parallel languages [13], including an evaluator based on the Maude system and a proof environment based on the PVS theorem prover. They have used this environment to compare several variants of the PLEXIL language.

6.2.7. Air Traffic Control

Participants: Gilles Dowek, Jeff Maddalon, Rick Butler, César Muñoz.

Gilles Dowek, Jeff Maddalon, Rick Butler, and César Muñoz have designed a prevention band algorithm and proved it correct [11].

6.2.8. Release, maintenance and documentation of the ssreflect extension

Participant: Assia Mahboubi.

In the context of her participation to the Mathematical Components project, Assia Mahboubi has pursued her contribution to the development, maintenance, documentation and distribution of the ssreflect extension. Version 1.2 of the ssreflect extension has been released in August 2009. The documentation has been adequately updated [17].

6.3. Study of Formalisms

6.3.1. Towards an implementation of the Implicit Calculus of Constructions

Participants: Bruno Barras, Bruno Bernardo.

Bruno Bernardo and Bruno Barras are working on an Implicit Calculus of Constructions with dependent sums (also known as Σ-types) and with decidable type inference. In this calculus, all the static information (types and proof objects), though it appears explicitly, is transparent and does not affect the computational behavior. Bruno Bernardo has already defined and studied an Implicit Calculus of Constructions with decidable typing and is now working on extending it with $\Sigma-$types, in order to have a more expressive language.

A problem in the metatheory of this extension has been uncovered: as in first-order logic with implicit existential quantifier, the subject-reduction property does not hold. However, this negative result does not affect more explicit versions of ICC, where introduction and elimination rules of the implicit quantifier are used explicitly in the proof derivation, but are considered implicit in the conversion rule. Next steps are to extend Alexandre Miquel's models based on coherence spaces in order to prove the consistency and the strong normalization property of the system.

6.3.2. Normalization and deduction modulo

Participants: Mathieu Boespflug, Denis Cousineau.

Mathieu Boespflug has been working on generalizing normalization by evaluation to term reduction modulo arbitrary rewrite rules in addition to beta-reduction. This allows for a cheap yet efficient implementation of a normalizer, as required in many proof assistants and also finds applications in partial evaluation. The implementation is cheap because most of the work is offloaded to an existing evaluator. His work has focused in particular on minimizing any overhead on beta-reduction caused by normalization by evaluation, showing that normalization by evaluation can reduce terms at nearly the same speed as the underlying evaluator.

He also wrote a full implementation of a type checker for proofs written in the $\lambda\Pi$-modulo calculus, called DEDUKTI. Version 1.0 of DEDUKTI was released in September. At this stage, the type checker works by translating the input terms into a functional program that can be compiled by the GHC Haskell compiler. Type checking the input terms is a side-effect of executing the obtained program. The translation makes essential use of untyped normalization by evaluation, a method for finding normal forms [5], [7].

Previously, Denis Cousineau worked on the property of strong normalization in logical frameworks where theories are expressed with rewrite rules. He defined, in particular, a semantic criterion not only correct but also complete for the property of strong normalization for theories expressed in Minimal Deduction modulo (a minimal version of Deduction modulo with the only two connectors $\Rightarrow$ and $\forall$) [18]. In 2009, Denis Cousineau has extended his results on a sound and complete semantic criterion for proof normalization to dependent types ($\lambda\Pi$-calculus modulo) [1], [8].

6.3.3. *Resolution and resolution modulo*

Participants: Gilles Dowek, Guillaume Burel.

Gilles Dowek has given a new formulation of Resolution modulo, called Polarized resolution modulo, that is both more general than Resolution modulo, because it includes polarized rewrite rules, and more restrictive, because all rewrite rules must be clausal [21]

Gilles Dowek has shown that simple type theory can be expressed as such a clausal rewrite system [22].

Polarized resolution modulo can be seen as a restriction of Equational resolution, that mixes clause restrictions and literal restrictions. Guillaume Burel and Gilles Dowek have compared this method with other restrictions of resolution and shown that as a consequence of Gödel second incompleteness theorem, Polarized resolution modulo is not an instance of Ordered resolution [20].

The text of the invited conference at LSFA 2007 of Gilles Dowek has been published. This text analyzes the history of the convergence of reduction-based methods and model-based methods in proof theory [2].

6.3.4. *Formalisation in deduction modulo*

Participant: Gilles Dowek.

Gilles Dowek has given a formulation in Deduction modulo of the system FA2 of J.-L. Krivine and M. Parigot. In particular, he has shown that one originality of this system is to express the specifications not as propositions, but in the congruence of the theory [23].

In a volume published in honor of Peter Andrews' birthday, Gilles Dowek has analyzed the history of higher-order Skolem theorem showing that this history illustrates a confrontation of two points of view on simple type theory: the logical and the theoretical points of view [3].

6.3.5. *Constructive mathematics*

Participant: Arnaud Spiwack.

Arnaud Spiwack (together with Thierry Coquand) has studied the notion of "finite set" in the setting of constructive mathematics. They describe how it naturally splits into several different properties. All of which are equivalent in ZFC. From a constructive point of view they have, however, quite different behaviours, and correspond algorithmically to different finite structures.

6.3.6. *Encoding rewriting strategies in lambda-calculi with patterns*

Participant: Germain Faure.

Germain Faure proposed an improvement to the pure pattern calculus: we claim that it is strictly more powerful to define the application of the match fail as the pure λ-term defining the boolean false instead of the identity function as it is done in the original version of the pure pattern calculus of Jay and Kesner. We show that using non algebraic patterns we are able to encode in a natural way any rewriting strategies as well as the branching construct "|" used in functional programming languages. We close the open question (raised by Cirstea and Kirchner) whether rewriting strategies can be directly encoded in λ-calculi with patterns. This work leads to a research report [14].

6.3.7. Semantics

Participants: Bruno Barras, Benjamin Werner.

Benjamin Werner and Bruno Barras work on various aspects of the set-theoretical models of Coq's type theory.

7. Contracts and Grants with Industry

7.1. INRIA Microsoft Research Joint Centre

TypiCal has a strong link with the INRIA-Microsoft Research joint centre, of which Benjamin Werner, Assia Mahboubi, Cyril Cohen, and Bruno Barras are also members.

7.2. ADT Coq

TypiCal, through its participation to the development of Coq is part of the ADT (Action de Développement Logiciel) Coq. It is a specific founding by INRIA. It involves people and teams that collaborate to the implementation of the Coq proof assistant. The involved teams are the following: TypiCal, ProVal, Marelle, and πr^2 from INRIA as well as the CPR team from CNAM.

7.3. Digitéo PASO

The PASO project *(Preuves, Interprétation abstraite, and Optimisation* cal properties of programs, arising in particular from the modeling of complex systems with critical security issues. It gathers computer scientists from CEA-LIST/MeASI, INRIA Saclay/Typical & LIX and specialists from Optimisation or Control theory from LIX/MeASI, INRIA Saclay/Maxplus & CMAP, and Supelec/L2S. The goal of this exploratory project is to cross-fertilise these fields, by applying advanced algorithms or techniques inspired by global optimization, by the analysis and identification of dynamical systems, or by zero-sum game theory, in order to improve the precision or the scalability of current methods in proof and static analysis. These applications coming from computer science turn out to raise new challenges for the applied mathematicians.

7.4. ANR Decert

Assia Mahboubi and Germain Faure are part of the ANR Decert *Décision certifiée* coordinated by Thomas Jensen in Rennes. The objective of the DECERT project is to design an architecture for cooperating decision procedures, with a particular emphasis on fragments of arithmetic, including bounded and unbounded arithmetic over the integers and the reals, and on their combination with other theories for data structures such as lists, arrays or sets. To ensure trust in the architecture, the decision procedures will either be proved correct inside a proof assistant or produce proof witnesses allowing external checkers to verify the validity of their answers.

7.5. ANR PSI

Assia Mahboubi and Germain Faure are part of the ANR PSI *Proof Search Interaction* coordinated by Stéphane Lengrand. The goal of the project is to understand how we can take into account a specific theory when elaborating proof search strategies, both at the level of proof theory and at the design of automated tools.

7.6. ARC Corias

Germain Faure (coordinator), Lisa Allali, Denis Cousineau, Gilles Dowek, Mathieu Boespflug are members of the ARC Corias *Conception et réalisation d'assistants à la preuve basés sur la super-déduction* in collaboration with the Pareo Team (INRIA Nancy Grand Est).

The project focusses on the development of an universal proof checker called Dedukti. It is based on the application of principles of (super)deduction modulo to type theory. The development of this tool requires (1) to obtain theoritical results for examples for the encoding of proofs from Coq (Calculus of Inductive Constructions formalism) in Dedukti (Lambda-calculus modulo formalism) (2) to develop implementation techniques for a successful scale-up.

7.7. European action: FORMATH

The FET-Open European project FORMATH about formalizing mathematics lead by Thierry Coquand in Göteborg is to start in 2010. It comprises researchers of INRIA, KUN Nijmegen, La Rioja, and Microsoft Research. Assia Mahboubi and Benjamin Werner are participating for TypiCal.

8. Dissemination

8.1. Animation of the scientific community

8.1.1. Organisation of Conferences and Workshops

Germain Faure and Assia Mahboubi organized an international workshop on the integration of SAT and SMT solvers into (Isabelle/)HOL on 10 and 11th September 2009 at LIX. People from Cambridge, Munich and Saclay attend this workshop.

8.1.2. Editorial charges

Assia Mahboubi serves in the program committee of the ITP-2010 (merge of the TPHOLs conference and ACL2 workshop) international conference.

Assia Mahboubi and Bruno Barras serve in the program committee of the JFLA 2010 national conference.

8.1.3. Committees

Assia Mahboubi has served in the committee "Commission de recrutement" for the "Maître de confèrence" position and "Chaire CNRS" in computer sciences at École Normale Supérieure de Lyon (ENS Lyon).

8.1.4. Referees

Germain Faure has served as referee for :

- the JFP'09 international journal,
- the CADE'09, FCT'09, FROCOS'09, RTA'09, and TLCA'09 international conferences and,
- the SMT'09 internation workshop.

Assia Mahboubi has served as referee for :

- the ICALP'09 and MKM'09 international conferences,
- the JFLA'10 national conference,
- the Annals of Mathematics and Artificial Intelligence, and the Journal of Formalized Mathematics and Computational Geometry, Theory and Applications international journals.

Bruno Barras refereed papers for JFLA'10 and Journal of Automated Reasoning.

Mathieu Boespflug refereed for the ICFP'09 international conference.

8.1.5. Visits

Assia Mahboubi has visited:

- Julien Narboux and Nicolas Magaud in December 2008 at Louis Pasteur University in Strasbourg. She has given a talk at the IGG team seminar.
- Georges Gonthier at Microsoft Research, Cambridge (UK) in January 2009.
- Marc Daumas and Erik Martin-Dorel in March 2009 at Perpignan Via Domitia University. She has given a talk at the ELIAUS team seminar.
- the LIP at École Normale Supérieure de Lyon, where she gave a talk at the Plume team seminar.
- the CSL group, as International Fellow of the Stanford Research Institute, Menlo Park, from mid-October to late December 2009.
- Nikolaj Bjorner and Leonardo de Moura at Microsoft Research Redmond, main developpers of the Z3 SMT solver, in December 2009. She gave a talk at the RiSE group seminar.

8.1.6. Conferences

Denis Cousineau has participated and given a talk at the following events or conferences:

- the French CNRS GDR-IM working groupe LAC at École Normale Supérieure of Lyon (France),
- the European workshop TYPES in Aussois (France) and,
- the international workshop PSTT at McGill University in Montréal (Canada).

Assia Mahboubi has participated and given a talk at the following events or conferences:

- the ADT meeting on Formalized libraries in December 2008 (Sophia Antipolis),
- the DECERT ANR project kick off meeting in January 2009 in Rennes,
- the JFLA national conference. She was invited to teach a course on the ssreflect extention,
- Microsoft Tech Fest in February 2009 (Redmond, U.S.). She was representing INRIA at the Microsoft INRIA Joint Centre booth.
- the ADT meeting on Automatization for the Coq system in March 2009 (Paris),
- Dagstuhl Seminar 09411 “Interation versus Automation, the two faces of deduction” in October 2009.

Germain Faure has participated and given a talk at the following events or conferences:

- ARC, ADT, and AEx 2009 day organised by INRIA, Bordeaux (France).
- the French CNRS GDR-IM working groupe LAC at École Normale Supérieure of Lyon, France).
- the ANR Decert *Décision certifiée* meeting, Paris, France.

Bruno Bernardo has attended TPHOLs 2009 and the Coq workshop in Munich, Germany (August 2009). He presented a poster at the Emerging Trends of TPHOLs’09.

Chantal Keller attended the TYPES´09 meeting (Aussois). She presented her Maters’s internship work about importing HOL-Light proofs into Coq.

Mathieu Boespflug attended LICS’09, NbE’09, and TYPES’09.

Arnaud Spiwack attended the TYPES’09 meeting (Aussois). He also took part in the Workshop “Formal Methods in Commutative Algebra” (Oberwolfach, Germany)

Bruno Barras participated to the following conferences:

- JFLA’09 from January 31 to February 3, Grenoble (France),
- the European workshop TYPES in Aussois (France),
- Workshop on Interactive Theorem Proving from August 24-25,Cambridge (UK),
- TPHOLs’09 and the First Coq Workshop from August 17-21, Munich (Germany).

Bruno Barras gave an invited talk at JFLA'09, and a regular talk at the Types workshop and the first Coq Workshop.

8.1.7. Popular science

Assia Mahboubi has given a popular science talk at the INRIA Saclay – Île-de- France popular science seminar "Unithé ou Café".

8.1.8. Other charges

Bruno Barras is consultant in formal methods at Trusted Labs, located in Versailles.

Benjamin Werner is a member of the scientific council of INRIA.

Benjamin Werner was vice-president of the recruiting comitee for joung researchers (CR) of INRIA-Saclay. He was member of the recruiting jury of the chaire d'excellence (young researcher, mixed reserch / teaching position) ENS-Cachan and INRIA.

Gilles Dowek is vice-head of the department of computer science of École Polytechnique.

Assia Mahboubi is representant of LIX researchers at the "Comité Enseignement-Recherche" of the teaching department of École polytechnique (DIX).

Assia Mahboubi is in charge of the TypiCal seminar.

Assia Mahboubi and Germain Faure have presented the LIX laboratory to the incoming students of the École Polytechnique in 2009.

Germain Faure is the coordinator of the INRIA ARC Corias that involves teams from INRIA Nancy Grand-Est, Bordeaux Grand-Ouest, and Saclay Île-de-France.

Denis Cousineau and Jean-Marc Notin are the webmasters of the Coq and TypiCal websites.

Denis Cousineau is the webmaster of Dedukti website.

8.2. Teaching

Assia Mahboubi and Benjamin Werner coadvise the thesis of Cyril Cohen. Benjamin Werner and Germain Faure coadvise the thesis of Chantal Keller. Benjamin Werner is thesis co-adviser of Arnaud Spiwack, Eric Biagoli, and François Garillot. Bruno Barras is the thesis advisor of Bruno Bernardo. Bruno Barras coadvises the thesis of Vincent Silès. Gilles Dowek is the thesis advisor of Mathieu Boespflug, Denis Cousineau, and Lisa Allali.

Assia Mahboubi, Bruno Barras, Gilles Dowek, and Benjamin Werner teach at the *Master Parisien de Recherche en Informatique*.

Germain Faure teaches at École Polytechnique.

Lisa Allali was a teaching assistant at the *Museum National d'Histoire Naturelle* until September 2009.

Vincent Siles and Arnaud Spiwack are teaching assistants at École Polytechnique.

Bruno Bernardo and Denis Cousineau were teaching assistants at the University Paris VII until September 2009.

Gilles Dowek is professor at École Polytechnique.

Mathieu Boespflug is a teaching assistant at École Polytechnique.

Benjamin Werner is part-time professor (professeur chargé de cours) at École Polytechnique. Since October 2009 he is vice-president of the computer science department, in charge of the 3rd and 4th study years.

Benjamin Werner teaches courses at Polytechnique for students of the 2nd and 3rd years (L3, M1) and gives a courses for master's students at MPRI (M2) in alternance with Gilles Dowek.

This year, Benjamin Werner gave a course on Coq at the VTSA'09 orgnized in Nancy by Stefan Merz for INRIA-Lorraine and Max-Planck Institut.

Assia Mahboubi has supervised the research master internship of Cyril Cohen (MPRI/ENS Cachan).

Assia Mahboubi has supervised the research internship of Salil Joshi (4th year student at IIT Delhi). This internship took place at the Inria Microsoft Research Joint Centre.

Assia Mahboubi has given a course on the ssreflect extension at the JFLA 2009 national conference.

Assia Mahboubi teaches at École Nationale Supérieure des Techniques Avancées (ENSTA).

Assia Mahboubi has taught a course on introduction to type-theory based proof assistant to the UNESCO-CIMPA summer school "Méthodes Effectives et Logiciels de la Logique et de l'Algèbre pour la Géomètrie Algébrique Réelle et la Cryptographie" at Yaoundé University (Cameroon).

9. Bibliography

Year Publications

Doctoral Dissertations and Habilitation Theses

[1] D. COUSINEAU. *Modèles et normalisation des preuves*, Ecole Polytechnique X, 12 2009, http://tel.archives-ouvertes.fr/tel-00433165/en/, Ph. D. Thesis.

Articles in International Peer-Reviewed Journal

[2] G. DOWEK. *On the convergence of reduction-based and model-based methods in proof theory*, in "Logic Journal of the Interest Group in Pure and Applied Logic", 2009.

[3] G. DOWEK. *Skolemization in Simple Type Theory: the Logical and the Theoretical Points of View*, in "Studies in Logic and the Foundations of Mathematics.", 2009.

International Peer-Reviewed Conference/Proceedings

[4] B. BARRAS. *Sets in Coq, Coq in Sets*, in "The first international Coq workshop", H. HERBELIN (editor), August 2009.

[5] M. BOESPFLUG. *Efficient normalization by evaluation*, in "2009 Workshop on Normalization by Evaluation, Los Angeles États-Unis d'Amérique", O. DANVY (editor), Olivier Danvy, 08 2009, http://hal.inria.fr/inria-00434283/en/.

[6] M. BOESPFLUG. *From Self-Interpreters to Normalization by Evaluation*, in "2009 Workshop on Normalization by Evaluation, Los Angeles États-Unis d'Amérique", O. DANVY (editor), Olivier Danvy, 08 2009, http://hal.inria.fr/inria-00434284/en/.

[7] M. BOESPFLUG. *Conversion by Evaluation*, in "Twelfth International Symposium on Practical Aspects of Declarative Languages, Madrid Espagne", 01 2010, http://hal.inria.fr/inria-00434282/en/.

[8] D. COUSINEAU. *Complete reducibility candidates*, in "Proof Search in Type Theory, Montréal Canada", 08 2009, http://hal.inria.fr/inria-00433159/en/.

[9] G. DOWEK, M. GABBAY, D. MULLIGAN. *Permissive nominal terms and their unification*, in "24th Italian Conference on Computational Logic", 2009.

[10] F. GARILLOT, G. GONTHIER, A. MAHBOUBI, L. RIDEAU. *Packaging Mathematical Structures*, in "Theorem Proving in Higher Order Logics, Munich Allemagne", T. NIPKOW, C. URBAN (editors), Lecture Notes in Computer Science, vol. 5674, Springer, 2009, http://hal.inria.fr/inria-00368403/en/.

[11] J. MADDALON, R. BUTLER, C. MUNOZ, G. DOWEK. *A Mathematical Analysis of Conflict Prevention Information*, in "9th AIAA Aviation Technology, Integration, and Operations Conference", 2009.

Scientific Books (or Scientific Book chapters)

[12] H. HERBELIN (editor). *The 1st Coq Workshop*, August 2009.

Research Reports

[13] G. DOWEK, C. MUNOZ, C. ROCHA. *Rewriting Logic Semantics of a Plan Execution Language*, n^{o} NASA TM-2009-215770, NASA, 2009, Under submission, Technical report.

[14] G. FAURE. *Encoding rewriting strategies in lambda-calculi with patterns*, n^{o} RR-7025, INRIA, 2009, http://hal.inria.fr/inria-00413178/en/, Research Report.

[15] G. FAURE. *Higher-order matching modulo (super)developements. Applications to second-order matching*, INRIA Saclay Île-de-France, 2009, http://hal.inria.fr/inria-00429978/en/, Research Report.

[16] G. FAURE, A. MIQUEL. *A Categorical Semantics for The Parallel Lambda-Calculus*, n^{o} RR-7063, INRIA, 2009, http://hal.inria.fr/inria-00424248/en/, Research Report.

[17] G. GONTHIER, A. MAHBOUBI. *A Small Scale Reflection Extension for the Coq system (v2.)*, n^{o} RR-6455, INRIA, 2009, http://hal.inria.fr/inria-00258384/en/, Research Report.

Other Publications

[18] D. COUSINEAU. *A completeness theorem for strong normalization in minimal deduction modulo*, 2009, http://hal.inria.fr/inria-00370379/en/, Under submission.

[19] D. COUSINEAU. *A semantic method to prove strong normalization from weak normalization*, 2009, http://hal.inria.fr/inria-00385520/en/, Under submission.

[20] G. DOWEK, G. BUREL. *How can we prove that a proof search method is not an instance of another?*, 2009, Under submission.

[21] G. DOWEK. *Polarized Resolution Modulo*, 2009, Under submission.

[22] G. DOWEK. *Simple Type Theory as a Clausal Theory*, 2009, Under submission.

[23] G. DOWEK. *Specifying programs with propositions and with congruences*, 2009, Under submission.