INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team MARELLE

Mathematics, Reasoning, and Software

Sophia Antipolis - Méditerranée

Theme : Programs, Verification and Proofs

Table of contents

1. Team

Research Scientist

Yves Bertot [Research scientist INRIA, HdR]
Benjamin Grégoire [Research scientist INRIA]
Laurence Rideau [Research scientist INRIA]
Loïc Pottier [Research scientist INRIA, HdR]
Laurent Théry [Research scientist INRIA]

Faculty Member

Frédérique Guilhot [Qualified teacher, *académie de Nice*]

Technical Staff

Anne Pacalet
Thomas Hutchinson
Evgeny Makarov
Benoit Razet

PhD Student

Sidi Ould Biha [Ph.D. student, advised by L. Théry]
Nicolas Julien [Ph.D. student, Teaching Assistant, advised by Y. Bertot]
Sylvain Heraud [Ph.D. student, advised by B. Grégoire]
Clément Hurlin [Ph.D. student, advised by M. Huisman]
Santiago Zanella [Ph.D. student, advised by Gilles Barthe]
Ioana Paşca [Ph.D. student, advised by Y. Bertot]
Tuan Minh Pham [Ph.D. student, advised by Y. Bertot]
Jorge Luis Sacchini [Ph.D. student, advised by B.Grégoire]
Michaël Armand [Ph.D. student, advised by L. Théry and B. Grégoire]

Administrative Assistant

Nathalie Bellesso [Administrative assistant]

2. Overall Objectives

2.1. Overall Objectives

We want to concentrate on the development of mathematical libraries for theorem proving tools. This objective contributes to two main areas of application: tools for mathematicians and correctness verification tools for software dealing with numerical computation.

In the short term, we aim for mathematical libraries that concern polynomials, algebra, group theory, floating point numbers, real numbers, big integers, probabilities and geometrical objects. In the long run, we think that this will involve any function that may be of use in embedded software for automatics or robotics (in what is called hybrid systems, systems that contain both software and physical components) and in cryptographical systems. We want to integrate these libraries in theorem proving tools because we believe they will become important tools for mathematical practice and for engineers who need to prove the correctness of their algorithms and software.

We believe that theorem proving tools are good tools to produce highly dependable software, because they provide a framework where algorithms and specifications can be studied uniformly and often provide means to automatically derive programs that are correct by construction.

Mathematical knowledge can also be made concrete in the form of decision procedures, often of the form of "satisfiability modulo theory" which can be connected to theorem proving tools in a way that preserves the trustability of the final results.

3. Scientific Foundations

3.1. Type theory and formalization of mathematics

The calculus of inductive constructions is a branch of type theory that serves as foundation for theorem proving tools, especially the Coq proof assistant. It is powerful enough to formalize complex mathematics, based on algebraic structures and operations. This is especially important as we want to produce proofs of logical properties for these algebraic structures, a goal that is only marginally addressed in most scientific computation systems.

The calculus of inductive constructions also makes it possible to write algorithms as recursive functional programs, which manipulate tree-like data structures. A third important characteristic of this calculus is that it is also a language for manipulating proofs. All this makes this calculus a tool of choice for our investigations. However, this language is still being improved and part of our work concerns these improvements.

3.2. Verification of scientific algorithms

To produce certified algorithms, we use the following approach: instead of attempting to prove properties of an existing program written in a conventional programming language such as C or Java, we produce new programs in the calculus of constructions whose correctness is an immediate consequence of their construction. This has several advantages. First, we work at a high level of abstraction, independently of the target implementation language. Second, we concentrate on specific characteristics of the algorithm, and abstract away from the rest (for instance, we abstract away from memory management or data implementation strategies). Thus, we are able to address more high-level mathematics and to express more general properties without being overwhelmed by implementation details.

However, this approach also presents a few drawbacks. For instance, the calculus of constructions usually imposes that recursive programs should explicitly terminate for all inputs. For some algorithms, we need to use advanced concepts (for instance, well-founded relations) to make the property of termination explicit, and proofs of correctness become especially difficult in this setting.

3.3. Programming language semantics

To bridge the gap between our high-level descriptions of algorithms and conventional programming languages, we also investigate the algorithms that occur when implementing programming languages, for instance algorithms that are used in a compiler or a static analysis tool. For these algorithms, we generally base our work on the semantic description of a language. The properties that we attempt to prove for an algorithm are, for example, that an optimization respects the meaning of programs or that the programs produced are free of some unwanted behavior. In practice, we rely on this study of programming language semantics to propose extensions to theorem proving tools or to participate in the verification that compilers for conventional programming languages are exempt of bugs.

3.4. Proof environments

We study how to improve mechanical tools for searching and verifying mathematical proofs so that they become practical for engineers and mathematicians to develop software and formal mathematical theories. There are two complementary objectives. The first is to improve the means of interaction between users and computers, so that the tools become usable by engineers, who have otherwise little interest in proof theory, and by mathematicians, who have little interest in programming or other kinds of formal constraints. The second objective is to make it easier to maintain large formal mathematical developments, so they can be re-used in a wide variety of contexts. Thus, we hope to increase the use of formal methods in software development, both by making it easier for beginners and by making it more efficient for expert users.

4. Application Domains

4.1. Certified scientific algorithms

For some applications, it is mandatory to build zero-default software. One way to reach this high level of reliability is to develop not only the program, but also a formal proof of its correctness. In the Marelle team, we are interested in certifying algorithms and programs for scientific computing. This is related to algorithms used in industry in the following respects:

- Arithmetical hardware in micro-processors,
- Arithmetical libraries in embedded software where precision is critical (global positioning, transportation, aeronautics),
- Verification of geometrical properties for robots (medical robotics),
- Verification of probabilities of breaking for cryptographic algorithms,
- Fault-tolerant and dependable systems.

5. New Results

5.1. Type theory and formalization of mathematics

5.1.1. Group theory

Participants: Georges Gonthier [Microsoft Research], Assia Mahboubi [project-team Typical], Laurence Rideau, Laurent Théry, Sidi Ould Biha.

We participate in the collaborative research agreement "Mathematical Components" with Microsoft Research. This project aims at evaluating the applicability of a new approach to mathematical proofs called "small-scale reflection", especially in the domain of finite group theory [2].

This year we have studied a methodology to define and combine algebraic structures, using dependent records, coercions and type inference, inside the Coq system. This alternative to telescopes in particular allows multiple inheritance, maximal sharing of notations and theories, and automated structure inference. Our methodology is robust enough to support a hierarchy comprising a broad variety of algebraic structures, from types with a choice operator to algebraically closed fields. Interfaces for the structures enjoy the handiness of a classical setting, without requiring any axiom. The library of algebraic strucutures we obtain is described in the paper [13].

We have also completed a formalisation of linear algebraic structures. This formalisation includes an algebraic hierarchy that covers vector spaces, algebra and modules over algebra. This hierarchy extend previous work on algebraic structures. We also developed theories of sub-structures and homomorphisms associated to these algebraic structures. These developments provide an infrastructure for the abstract theory of representation and character of finite groups. It was the subject of a paper and an oral presentation at the Mathematical Knowledge Management 2009 conference [18].

5.1.2. On Bernstein coefficients

Participants: Assia Mahboubi [Project-team Typical, INRIA Saclay – Île-de-France], Yves Bertot.

We re-visited a proof that a polynomial with a single change of sign among its coefficients has exactly one real root between 0 and positive infinity. The new proof is more systematic and constructive, it relies on rational numbers in a clearer way and is adapted to the ssreflect style also advocated in our group theory work. The corollary that a polynomial with a single change of sign among its Bernstein coefficients for a given interval has a single root inside this interval is the next step. In the long run, this should contribute to a formally verified implementation of cylindrical algebraic decomposition.

5.1.3. Formalising Geometric Algebras

Participants: Laurent Fuchs [Université de Poitiers], Laurent Théry.

As part of the Galapagos project, we further improve our formalisation of Geometric algebras. We have implemented and proved correct the join and meet operations of the Cayley-Grassman algebras and verified a factorisation algorithm.

5.1.4. Ptolemy's theorem

Participants: Yves Bertot, Frédérique Guilhot, Tuan Minh Pham.

We have submitted a paper describing our work on Ptolemy's theorem, a theorem about cocyclic points where oriented triangles play a role.

5.1.5. Dependent elimination and type-based termination

Participants: Bruno Barras [Project-team Typical, INRIA Saclay – Île-de-France], Pierre Corbineau [Université Joseph Fourier], Hugo Herbelin [Project-team $\pi\,\mathrm{r}^2$, INRIA Paris-Rocquencourt], Benjamin Werner [Project-team Typical, INRIA Saclay – Île-de-France], Benjamin Grégoire, Jorge Luis Sacchini.

We worked on extending the elimination rule of the Calculus of Inductive Constructions (CIC) to automatically perform inversion reasoning on dependent data structures. The results were published in [6].

We also worked on extending the type system of CIC to perform type-based termination. In particular, we focused on proving metatheoretical properties of the system, namely, strong normalization and decidability of type checking. We have proved these properties for a subsystem without universes, and we are currently working on the proofs for the full system.

5.1.6. Proof of the Java Bytecode Verifier

Participants: Benjamin Grégoire, David Pichardie [Project-team Celtique, IRISA].

We have developed a formal proof of correctness for the Java Bytecode Verifier (BCV). This work was done on top of Bicolano, the formal semantic of the Java bytecode developed by David Pichardie. The algorithm can be executed in Coq, and we plan to extract it to get a certified BCV executable in Ocaml.

5.2. Verification of scientific algorithms

5.2.1. Termination of Delaunay triangulation by edge flipping

Participants: Jean-François Dufourd [Université de Strasbourg], Yves Bertot.

An algorithm to produce planar Delaunay triangulations has been formally described in the framework of the ANR Galapagos project. In particular, we showed that the termination of the algorithm could be proved with the help of a systematic description of finite sets.

5.2.2. Co-recursion and real numbers

Participants: Yves Bertot, Nicolas Julien, Ioana Paşca.

The traditional understanding that real numbers are fractional numbers with an infinite sequence of digits after the decimal point can be modeled using infinite streams of digits, a special case of co-inductive data-types. The main work of this year was to optimize a generic approach to computations based on Newton's algorithm, for instance for the square-root function. Part of this work was published in [17].

5.2.3. Regularity of interval matrices

Participants: Yves Bertot, Ioana Paşca.

We are studying an article by Rex and Rohn that gives efficient sufficient conditions for regularity of matrices with interval coefficients. Although it is in a preliminary stage, we can foresee that it will rely on our previous work on formalizing linear algebra.

Like our previous work on Kantorovitch's theorem that lead to our formalization of Newton's method, this topic was proposed to us by colleagues from the COPRIN team, who are more involved in robotics. In the long run we expect that a formal description of the convergence theorems makes it possible to propose new tools for the verification of controling software in this domain.

5.2.4. Certificate translation for optimizing compiler

Participants: Gilles Barthe [IMDEA Madrid], Benjamin Grégoire, Sylvain Heraud, Cesar Kunz [IMDEA Madrid], Anne Pacalet.

In a Proof Carrying Code environment, certificate generation remains an open problem. Certifying compilers can automatically produce certificates but are mostly restricted to basic safety properties. Certificate translation is an alternative method that transforms certificates of source programs into certificates of compiled programs. In an earlier work we have developed the theory of certificate translation. This years we have developed an implementation. It was the subject of a paper at International Conference on Formal Engineering Methods [7], in particular we study the impact of certificate translation on the size of certificates.

5.2.5. Certifying code generation

Participants: Benjamin Grégoire, Jan-Olaf Blech [Verimag Grenoble].

In [12], we have presented an approach to guarantee the correctness of compiler transformations with respect to a formal notion of correctness. We certify the results of each compilation run. With the help of a compiler generated certificate and a certificate checker, we verify the results of each compilation run automatically. Thus, we ensure the correctness of the compilation run without having to look at concrete compilation algorithms. A journal version has been submitted to Formal Methods in System Design.

5.2.6. Formally verified structural abstract interpretation

Participant: Yves Bertot.

A paper describing a formal study of abstract interpretation has been published in the lecture notes of a summer school [10].

5.2.7. Properties of Gene networks

Participants: Yves Bertot, Johan Segura, Adrien Richard [Laboratoire CNRS I3S, Université de Nice-Sophia Antipolis].

Biologists often try to predict the dynamic behavior of complex biological systems composed of several genes by studying interactions between some of these genes. For instance, a gene may be known to activate or inhibit the expression of another one. An abstract *interaction graph*, with positive and negative edges, is then used as a tool to model activatory and inhibitory effects between genes. Another abstract model relies on notions of finite-state automata. We studied the relations between some aspects of the two models. This year we concentrated on an article by Shih and Dong on sufficient conditions for the existence of a single fixed point, in the case where each gene can only have two states.

An article on the work in the same domain from previous years has been submitted to an international conference.

5.3. Tools for proof environments

5.3.1. Gröbner bases

Participants: Loïc Pottier, Benjamin Grégoire, Laurent Théry.

To prove automatically polynomial equalities in Coq via the nullstellensatz theorem of Hilbert, we wrote a tactic that computes Gröbner bases, called gb, and tested it extensively on geometrical examples. After that, we remarked that the computation of the whole Gröbner basis was not necessary. We obtained a much more efficient tactic, which is able to prove state-of-the art geometrical theorems, like Pappus and Desargues. This work is part of our ANR funded Galapagos project.

5.3.2. Coq normalization

Participants: Loïc Pottier, Benjamin Grégoire.

In order to increase speed of computations in Coq, we reused a work on extraction done in 1998, to produce ocaml code from Coq terms. Using native compilation, this code produces the normal form of the term. The problem is then to lift the result in Coq, which has been done using the techniques developped in the VM of Coq. Experimental times on several big examples were good, not so far from times obtained by native compilation of extracted terms. We then concentrated on terms that normalize on pure inductive terms, as only needed by reflexive techniques, and began to develop a compiler from Coq to C. Experiments on examples are promising: computation times can be divided by 4.

5.3.3. Geometric proof environment

Participants: Yves Bertot, Tuan Minh Pham, Frédérique Guilhot.

To develop interactive proofs in geometry with a good feedback on the geometric figures, we developed a hybrid proof environment including the GeoGebra tool, a mainstream dynamic geometry tool, Coq, and Pcoq. This hybrid system makes it easy to integrate many functionalities for efficient man-machine interaction. In particular, we want to study a facility for fast use of geometric theorems that reuses ideas of proof-by-pointing and automatic computation of dependances as in Geoview, a previous experiment developed in our team.

5.3.4. Satisfiability in Coq

Participants: Michaël Armand, Benjamin Grégoire, Laurent Théry.

We have integrated state of the art procedures to perform boolean satisfiability checking inside Coq. We followed two different paths. In the first one, we directly programmed the standard DPLL algorithm inside Coq. This gave us a way to perform satisfiability by reflection. In the second one, we took advantage of the capability of a Sat solver like `zchaff` or `minisat` to produce a log of their run in terms of resolution traces and we have implemented a certified version of a trace checker. We tested boolean Gröbner bases on Sat test problems, because certificates are easy to obtain with this tool. Unfortunately, this technique is not powerful enough on the test corpus.

5.3.5. WP plugin for Frama-C framework

Participants: Anne Pacalet, Guillaume Claret.

Weakest Precondition computation is a way to provide proof obligations that ensures that some given properties of programs hold. This is an old technique, but it is difficult to apply on the C language. The 2009-2011 objective is to develop, together with the CEA, a WP plugin for the C static analysis framework Frama-C. Our aim is to provide several memory models in order to adapt the abstraction level of the verification. During 2009, we managed to develop a generic engine and two memory models : one that is very abstract but only applies to few programs, and another low level one that can apply only to small sequences. Both are correct, but the challenge is then either to find an intermediate model or to make the two existing models collaborate, in order to be able to process real applications. In parallel to the tool development, we tried to formalize some parts of the models using the COQ proof assistant to check the correctness of new ideas.

5.3.6. Certification of cryptographic primitives

Participants: Gilles Barthe [IMDEA Madrid], Benjamin Grégoire, Daniel Hedin [IMDEA Madrid], Sylvain Heraud, Santiago Zanella.

CertiCrypt is an automated framework to construct and verify security proofs of cryptographic systems in the computational model. It was the subject of two papers, an overview at Principles of Programming Langages [9] and a paper on Full-Domain Hash Signature at Security and Privacy [8]. We also worked on Zero-Knoweledge Protocols, Full Domain Hash (FDH), Optimal Asymmetric Encryption Padding (OAEP IND-CCA2) and a hash function based on elliptic curves.

In [16], we extended our earlier work on specification of protcols of classes. We deal with a variant of generic classes and multithreaded classes. Because little support currently exists to help writing method contracts, our technique helps programmers to check their contracts early in the development process.

In [5], we extend our earlier work on specifications of iterators with separation logic contracts. We present examples of iterator clients and implementations and proofs that they adhere to the iterator protocol.

In [14], we describe an algorithm to disprove entailment between separation logic formulas. This is of interest wherever entailment checking is performed (such as in program verifiers). Our algorithm has been implemented and verified in Coq.

In [15], we show how, given a program and its separation logic proof, one can parallelize and optimize this program and transform its proof simultaneously to obtain a proven parallelized and optimized program (using the éterlou program, in the tom rewriting framework). A longer version of this publication appeared in technical report 6806 [21].

6. Other Grants and Activities

6.1. National initiatives

- We participated in the national contract A3PAT, which started on Dec. 1st 2005. Other participants in this contract are CEDRIC-CNAM (Evry), LABRI (Bordeaux), and LRI (Orsay). The objective of this contract is to study the possible combination of the rewriting engine Cime and the Coq system, especially in the verification that recursive algorithms do terminate. This contract was terminated in June 2009.
- We participate in the common laboratory between INRIA and Microsoft Research, in the Collaborative research actions “Mathematical components” and “Secure Distributed Computations and their Proofs”. Other participants in the first collaboration are the INRIA project-teams TYPICAL and PROVAL. The goals are to study finite group theory and efficient arithmetics. In the second collaboration, other participants are the INRIA teams INDES (formerly MIMOSA) and MOSCOVA. We focus on formal proofs for computational Cryptography.
- We lead the national contract Galapagos, which started on Nov. 19th 2007. Other participants in this contract are the universities of Strasbourg and Poitiers, the ENSIEE in Evry and the Ecole Normale Supérieure in Lyon. The objective of this contract is to study the formal description of geometric concepts and algorithms.
- We participate to the ANR SCALP, which started on January 1st, 2008. Other participants in this contract are DCS-Verimag (Grenoble), Plume-LIP (Lyon), Proval-LRI (Orsay), CPR-Cédric (Cnam, Paris). In this project we focus on the formalization of Cryptography.
- We participate to the ANR DeCert, which started on January 2009. Other participants are CEA List (Paris), LORIA-INRIA (Nancy), Celtique-IRISA (Rennes), Proval-LRI (Orsay), Typical-INRIA Futurs, Systerel (Aix-en-provence). The objective of the DeCert project is to design an architecture for cooperating decision procedures. 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.
- We collaborate with the CEA to develop Frama-C which is a suite of tools dedicated to the analysis of the source code of software written in C.

6.2. European initiatives

- Together with the Universities of Chalmers (Sweden), Nijmegen (the Netherlands), and La Rioja (Spain), we applied for funding from the European Community. The project, named Formath (Formalization of Mathematics) has been accepted in the ICT program (grant agreement number 243847).

  In this project, we will concentrate on developing mathematical libraries for algebra, linear algebra, and algebraic topology.

7. Dissemination

7.1. Conference and workshop attendance, travel

Santiago Zanella attented the conference POPL in Savannah, Georgia (USA), in January, where he presented a paper.

Sylvain Heraud, and Santiago Zanella participate to the workshop CoSyProofs (Computational and Symbolic Proofs of Security) (Highashi Izu - Japan), in April. Santiago Zannella gave a talk.

Yves Bertot gave an invited lecture at the conference JCSSE in Phuket, Thailand, in May.

Ioana Paşca, Loïc Pottier, Laurence Rideau, and Jorge Luis Sacchini attended the conference Types'09, in Aussois, in May.

Santiago Zanella attended the IEEE symposium on Security and Privacy, in Oakland (California, USA), in May where he pressented a paper.

Sylvain Heraud participated and gave an talk at the Summer School EJCP (Ecole des Jeunes Chercheurs en Programmation), in Rennes and Dinard, in June.

Sylvain Heraud visited IMDEA Madrid between June and July.

Jorge Luis Sacchini visited the Project-Team Typical (INRIA Saclay – Île-de-France) between June and September.

Laurence Rideau and Sidi Ould Biha attented the MKM'09 conference in Grand Bend (Ontario, Canada),in July.

Nicolas Julien, Ioana Paşca, and Laurent Théry attended the TPHOLs conference (Theorem Proving in Higher Order Logics), in Munich, in August, where they presented two papers.

Laurent Théry attended the workshop on Interactive Theorem Proving at Cambridge (UK), in August.

Yves Bertot gave a course at the Coq Summer School at the University of Tsinghua (Beijing, China), in August.

Sylvain Heraud gave a talk at the Sophia-Antipolis Formal Analysis Group (SAFA), in September.

Yves Bertot attended the conference Synasc'09, in Timisoara (Roumania) in September where he presented a paper.

Sylvain Heraud participated and gave an talk at Summer School On Provable Security, in Barcelona (Spain), in September.

Benjamin Grégoire and Sylvain Heraud travelled to IMDEA (Madrid) to collaborate with Gilles Barthe, in September.

Michaël Armand, Benjamin Grégoire, and Laurent Théry participated to the workshop Integrating SAT and SMT Solvers with Isabelle/HOL in Paris. Michaël Armand gave a talk on integrating SAT in Coq, in September.

Laurent Théry attended the seminarium "Interaction versus Automation", in Dagstuhl (Germany), in October.

Ioana Paşca took part in the annual meeting on arithmetic, computer science, and mathematics (RAIM–Rencontres annuelles sur l'Arithmétique l'Informatique et les Mathématiques) in Lyon, and gave a talk on exact real arithmetics in theorem provers.

Laurent Théry gave an invited talk at the ARENAIRE seminar at Ens Lyon in October.

Michaël Armand, attended the VTSA summer School, in Nancy, in October.

Yves Bertot and Laurence Rideau attended the MAP'09 Conference, in Monastir (Tunisia), in December.

Sylvain Heraud, Benjamin Grégoire, and Santiago Zanella took part to meetings of the SCALP ANR project in Paris (February), Madrid (May), and Orsay (October).

Michaël Armand, Benjamin Grégoire, Loïc Pottier , and Laurent Théry took part in meetings of the DECERT ANR project, in June and in November.

Yves Bertot, Benjamin Grégoire, Thomas Hutchinson and Loïc Pottier participated to the meetings of ADT Coq (Action de Développement Technologique) in Palaiseau (January), Paris (March and June), and Bordeaux (October).

Yves Bertot, Ioana Paşca, Laurence Rideau, and Laurent Théry regularly participated to meetings in Saclay for the common laboratory INRIA/Microsoft.

7.2. Leadership within scientific community

- Yves Bertot was a co-editor of the book "From Semantics to Computer Science, essay in honour of Gilles Kahn", published by Cambridge University Press in October.
- Yves Bertot was a member of the program committee for the conference FICS (Fixpoints in Computer Science) and for the Coq workshop.
- Laurence Rideau was a member of the program committee for the conference MKM (Mathematical Knowledge Management).
- Laurent Théry was a member of the program committee for the conference TPHOLs (Theorem Proving in Higher Order Logics).
- Laurent Théry was the program chair for the workshop PLMMS (Programming Languages for Mechanized Mathematics Systems).
- Yves Bertot gave talks at the School for young researchers in programming on *introduction to type theory* and *Coq in a Hurry*.
- Project members reviewed papers for the journals JAR (Journal of Automated Reasoning), TOMS (Transactions on Mathematical Software), TSI (Technique et Science Informatique), ESL (Embedded Systems Letters) and for the conferences MKM (Mathematical Knowledge Management), POPL (Principles of Programming Languages), JFLA (Journées Francophones des Langages Applicatifs), TPHOLs (Theorem Proving in Higher Order Logics), PLMMS (Programming Languages for Mechanized Mathematics Systems).

7.3. Miscellaneous

- Yves Bertot was a member of the habilitation jury for Pierre Crégut at the University of Rennes 1.
- Laurent Théry was the external examiner for the PhD of Clelia Lomuto at the University of Florence.
- Benjamin Grégoire was member of the Ph.D. jury of Cesar Kunz at Ecole Nationale Supérieure des Mines de Paris, in February.

7.4. Supervision of Ph.D. projects

- Laurent Théry supervises the Ph.D. project of Sidi Ould Biha, which started on 2006, Sept. 1st, with funding from the INRIA-Microsoft research common laboratory.

- Yves Bertot supervises the Ph.D. project of Nicolas Julien, which started on Oct. 1st, 2006, with funding from the French ministry of research and a teaching assistant grant.
- Yves Bertot supervises the Ph.D. project of Ioana Paşca, which started on Oct. 1st, 2007, with funding from the French ministry of research.
- Yves Bertot supervises the Ph.D. project of Tuan Minh Pham, which started on March 1st, 2008, with funding from the ANR GALAPAGOS project.
- Benjamin Grégoire supervises the Ph.D. project of Sylvain Heraud, which started on September 1st, 2008, with funding from the ANR SCALP project.
- Benjamin Grégoire co-supervises the Ph.D. project of Jorge-Luis Sacchini, which started on November 1st, 2007, with funding from the Mobius european project.
- Benjamin Grégoire and Laurent Théry co-supervises the Ph.D. project of Michaël Armand which started on October 1st, 2009, with funding from the ANR DECERT project.
- Gilles Barthe supervised the Ph.D. project of César Kunz started which started in September 2005, and was defended on February 3rd, 2009.
- Gilles Barthe supervises the Ph.D. project of Santiago Zanella which started in June 2006, with funding from the INRIA/Microsoft research laboratory.
- Marieke Huisman supervised the Ph.D. project of Clément Hurlin which started on September 1st, 2006, with funding from Mobius european project and was defended on September 14th, 2009.

7.5. Teaching

Yves Bertot *Sémantique des langages de programmation I* (Programming language semantics I), 1st year Master (18 hours), University of Nice. *Sémantique des langages de programmation, techniques avancées* (Programming language semantics, advanced techniques), 1st year Master, special cursus at University of Nice (pensionnaires de l'école normale supérieure).

Nicolas Julien *Introduction à la programmation fonctionnelle*, University of Nice (48 hours), *Systèmes et réseaux*, University of Nice (22 hours), *Algorithmique et informatique théorique*, University of Nice (16 hours).

Sylvain Heraud *Introduction à la programmation fonctionnelle*, University of Nice (48 hours).

Ioana Paşca *Travaux pratiques de Mathématiques Discrètes*, (Lab. sessions of discrete mathematics), 1st year (84 hours), University of Nice.

Loïc Pottier *Sémantique des langages de programmation I* (Programming language semantics I), 1st year Master (50 hours), University of Nice, *Preuves formelles* (Formal proofs), 2nd year Master (20 hours), University of Nice, *Preuves formelles* (Formal proofs), 2nd year Master (3 hours), University of Aix-Marseille.

Laurent Théry *Proof Mechanization*, 2nd year Master, University of Marseilles (3 hours). *Introduction to Coq*, École des Mines de Paris, (3 hours).

Benjamin Grégoire *Vérification et Sécurité*, 2nd year Master, Ecole Polytechnique de l'Université de Nice Sophia-Antipolis.

8. Bibliography

Major publications by the team in recent years

[1] Y. BERTOT, P. CASTÉRAN. *Interactive Theorem Proving and Program Development, Coq'Art:the Calculus of Inductive Constructions*, Springer-Verlag, 2004.

[2] G. GONTHIER, A. MAHBOUBI, L. RIDEAU, E. TASSI, L. THÉRY. *A Modular Formalisation of Finite Group Theory*, in "Proceedings of the 20th International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2007)", K. SCHNEIDER, J. BRANDT (editors), LNCS, vol. 4732, Springer-Verlag, September 2007, p. 86-101, http://hal.inria.fr/inria-00139131.

[3] L. THÉRY. *A Machine-Checked Implementation of Buchberger's Algorithm*, in "Journal of Automated Reasoning", vol. 26, 2001, p. 107–137.

Year Publications

Doctoral Dissertations and Habilitation Theses

[4] C. HURLIN. *Specification and Verification of Multithreaded Object-Oriented Programs with Separation Logic*, Université Nice - Sophia Antipolis, September 2009, http://www-sop.inria.fr/everest/Clement.Hurlin/publis/these.pdf, Ph. D. Thesis.

Articles in International Peer-Reviewed Journal

[5] C. HAACK, C. HURLIN. *Resource Usage Protocols for Iterators*, in "Journal of Object Technology", vol. 8, n° 4, 2009, http://www.jot.fm/issues/issue_2009_06/article3/index.html, This is an extended version of a paper that appeared in the IWACO'08 workshop.

International Peer-Reviewed Conference/Proceedings

[6] B. BARRAS, P. CORBINEAU, B. GRÉGOIRE, H. HERBELIN, J. L. SACCHINI. *A New Elimination Rule for the Calculus of Inductive Constructions*, in "Types for proofs and programs 2008", S. BERARDI, F. DAMIANI, U. DE'LIGUORO (editors), Lecture Notes in Computer Science, vol. 5497, Springer, 2009, p. 32-48.

[7] G. BARTHE, B. GRÉGOIRE, S. HERAUD, C. KUNZ, A. PACALET. *Implementing a direct method for certificate translation*, in "International Conference on Formal Engineering Methods, ICFEM 2009", Lectures Notes in Computer Science, 2009, To appear.

[8] G. BARTHE, B. GRÉGOIRE, F. OLMEDO, S. ZANELLA BÉGUELIN. *Formally Certifying the Security of Digital Signature Schemes*, in "30th IEEE Symposium on Security and Privacy, S&P 2009", IEEE, 2009.

[9] G. BARTHE, B. GRÉGOIRE, S. ZANELLA BÉGUELIN. *Formal Certification of Code-Based Cryptographic Proofs*, in "36th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2009", ACM, 2009, p. 90–101, http://dx.doi.org/10.1145/1480881.1480894.

[10] Y. BERTOT. *Structural abstract interpretation, A formal study in Coq*, in "Language Engineering and Rigorous Software Development, International LerNet ALFA Summer School 2008, revised tutorial lectures", A. BOVE, L. S. BARBOSA, A. PARDO, J. S. PINTO (editors), Lecture Notes in Computer Science, vol. 5520, Springer, 2009, p. 153-194, http://hal.inria.fr/inria-00329572/.

[11] Y. BERTOT, E. KOMENDANTSKAYA. *Using Structural Recursion for Corecursion*, in "Types for proofs and programs 2008", S. BERARDI, F. DAMIANI, U. DE'LIGUORO (editors), Lecture Notes in Computer Science, vol. 5497, Springer, 2009, p. 220-236, http://hal.inria.fr/inria-00322331.

[12] J. O. BLECH, B. GRÉGOIRE. *Using Checker Predicates in Certifying Code Generation*, in "Proceedings of the Workshop Compiler Optimization meets Compiler Verification (COCV 2009)", ENTCS, 2009, To appear.

[13] 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/, G.: Mathematics of Computing/G.4: MATHEMATICAL SOFTWARE, I.: Computing Methodologies/I.1: SYMBOLIC AND ALGEBRAIC MANIPULATION/I.1.0: General.

[14] C. HURLIN, F. BOBOT, A. J. SUMMERS. *Size Does Matter : Two Certified Abstractions to Disprove Entailment in Intuitionistic and Classical Separation Logic*, in "International Workshop on Aliasing, Confinement and Ownership in object-oriented programming (IWACO'09)", July 2009, http://www-sop.inria.fr/everest/Clement.Hurlin/publis/iwaco09.pdf, Coq proofs: disprove.tgz.

[15] C. HURLIN. *Automatic Parallelization and Optimization of Programs by Proof Rewriting (or Automatic Parallelization with Separation Logic!)*, in "Static Analysis Symposium (SAS'09)", Lecture Notes in Computer Science, vol. 5673, Springer-Verlag, August 2009, p. 52–68, http://www-sop.inria.fr/everest/Clement.Hurlin/publis/eterlou.pdf, A longer version appeared as technical report 6806 from INRIA. The implementation is available..

[16] C. HURLIN. *Specifying and Checking Protocols of Multithreaded Classes*, in "ACM Symposium on Applied Computing (SAC'09)", ACM, March 2009, p. 587–592, http://www-sop.inria.fr/everest/Clement.Hurlin/publis/sac09.pdf, Additional material (examples, detailed statistics, and implementation) is available: pyrolobus..

[17] N. JULIEN, I. PASCA. *Formal Verification of Exact Computations Using Newton's Method*, in "Proceedings of the 22nd International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2009)", LNCS, vol. 5674, Springer, August 2009, p. 408-423, http://hal.inria.fr/inria-00369511/en/.

[18] S. OULD BIHA. *Finite groups representation theory with Coq*, in "Mathematical Knowledge Management", LNAI, vol. 5625, Springer, July 2009, p. 438-452, http://hal.inria.fr/inria-00377431.

Scientific Books (or Scientific Book chapters)

[19] Y. BERTOT. *Theorem proving support in programming language semantics*, in "From Semantics to Computer Science, essays in Honour of Gilles Kahn", Y. BERTOT, G. HUET, J.-J. LÉVY, G. PLOTKIN (editors), Cambridge University Press, 2009, p. 337–361, http://hal.inria.fr/inria-00160309/.

Books or Proceedings Editing

[20] Y. BERTOT, G. HUET, J.-J. LÉVY, G. PLOTKIN (editors). *From Semantics to Computer Science, essays in Honour of Gilles Kahn*, Cambridge University Press, 2009.

Research Reports

[21] C. HURLIN. *Automatic Parallelization and Optimization of Programs by Proof Rewriting (or Automatic Parallelization with Separation Logic!)*, n^{o} 6806, INRIA Sophia-Antipolis – Méditerrannée, June 2009, Technical report.