INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team MARELLE

Mathematics, Reasoning, and Software

Sophia Antipolis - Méditerranée

THEME SYM

Table of contents

1. Team

Research Scientist

Yves Bertot [Research scientist INRIA, HdR]
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*]

PhD Student

Sidi Ould Biha [Phd. student, advised by L. Théry]
Nicolas Julien [Teaching Assistant, advised by Y. Bertot]
Ioana Pasca [Phd. student, advised by Y. Bertot]
Tuan Minh Pham [Phd. student, arrived on March 3rd, advised by Yves Bertot]

Post-Doctoral Fellow

Vladimir Komendantsky [until October 2008]
Ekaterina Komendantskaya [until September 2008]

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 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). 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.

In 2007, we focused on algorithms concerning matrices, group theory, multi-variate analysis, and program analyses.

3. Scientific Foundations

3.1. Type theory and formalization of mathematics

Keywords: *Coq*, *formalization*, *mathematics*, *type theory*.

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

Keywords: *Coq*, *algorithms*, *certification*.

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

Keywords: *Coq*, *programming languages*, *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

Keywords: *Coq*, *environments*, *man-machine interface*, *proofs*.

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),
- Fault-tolerant and dependable systems.

5. New Results

5.1. Type theory and formalization of mathematics

5.1.1. Imperative Arrays in Coq

Participants: Benjamin Grégoire [project-team EVEREST], Laurent Théry.

The programming language inside the Coq system is purely functional. In particular, this means that the arrays that are available in Coq are not the usual ones but functional arrays with no side-effect. Recently, Arnaud Spiwack introduced the retro-knowledge paradigm in Coq. It gives the opportunity to import imperative features inside Coq as long as they can be encapsulated inside a purely functional interface. We have experimented with a version of imperative arrays with such a functional interface. Our results show that an impressive speed-up is gained, especially in applications where look-ups are much more frequent than updates.

5.1.2. Modelling and extracting general recursive functions

Participants: Yves Bertot, Vladimir Komendantsky.

As part of our development of a certified extraction of general recursive functions from Coq specifications, we implemented a formalism of complete preorders and continuous functions. We have then completed this formal development into an extension of the Coq system, which derives a formal description of potentially non terminating functions from their description as recursive functions, together with various theorem to help reasoning on these functions. This tool corresponds to the `Function` tool, but for non-terminating functions. This work is published in [8].

5.1.3. Proofs by state exploration

Participant: Laurent Théry.

The recent introduction in Coq of native integers makes it possible to implement a very compact representation of finite sets in a purely functional setting. To illustrate this, we have formally proved that a maximun number of 11 moves is needed to solve any configuration of the Mini Rubik (the 2x2x2 version of the famous Rubik's Cube). This amounts to exhaustively exploring all the possible configurations of the MiniRubik, i.e. a finite set of 3674160 elements.

5.1.4. 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 completed the basic notions of group theory with nilpotent groups and their properties. From June, we have started formalising the first introductory chapter of Bender and Glauberman's book that describes the proof of Feit-Thompson theorem.

In parallel, we have intensified our effort towards linear algebra as it is necessary for the mathematical theory known as *character theory*. First, the library of univariate polynomials has been completed with standard operation like pseudo-division. Second, a proper formalisation of vector spaces has been developed. As a basic decision procedure for some of the operations on vector spaces, gauss elimination has also been formalised. This work leads to a cleaner proof of Maschke's theorem.

As a fundamental tool to our investigations on linear algebra, we also studied some aspects of matrix and determinant computations. This lead us to streamlining a framework for iterated operators (big operators), like repeated sums, or products, where the index ranges over a finite domain. The library of operators we obtain is described in the paper [6].

5.1.5. Formalising Geometric Algebras

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

Geometric algebras are effective tools to reason in a very abstract way about geometrical problems. We have started a formalisation of these algebras in Coq starting from the basic Grassman algebra. This work is part of the Galapagos project.

5.1.6. Ptolemy's theorem

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

In the context of the Galapagos ANR project, we studied how Ptolemy's theorem could be formalised on top of the formal development of High School Geometry. This theorem states that for a convex quadrangle whose four vertices lie on a circle, the product of lengths of the diagonals is equal to the sum of products of the two pairs of opposite sides. This theorem required that we add notions of orientation to the formalisation of geometry. Ongoing work concentrates on the notion of orientation, through the formalisation of other theorems where this notion plays a role.

5.2. Verification of scientific algorithms

5.2.1. Co-recursion and real numbers

Participants: Yves Bertot, Nicolas Julien, Ioana Pasca.

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. For algorithms to have a good behavior, we need to work in a generalized framework where digits can be positive or negative. The main work of this year was to optimize division and the construction of streams for rational numbers, to study a generic approach to computations based on Newton's algorithm, for instance for the square-root function, and to start the implementation of trigonometric functions. Part of this work was published in the paper [10].

5.2.2. The Kantorovitch theorem

Participants: Yves Bertot, Ioana Pasca.

Newton's method is frequently used to find the roots of continuous, twice differentiable multivariate functions. In previous work we studied the formal proof of a theorem stating sufficient conditions for this method to converge. This year, we studied how this method could be included in the library for exact real number computation. This required to re-design the proof to incorporate rounding operations.

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. Part of this work is described in the collective paper [6].

5.2.3. Formally verified compilation

Participants: Xavier Leroy [project-team GALLIUM], Laurence Rideau, Bernard Serpette.

In previous years, we worked on the formal description of an algorithm for the parallel move from collections of registers to collections of registers. This work appeared this year in a journal article [4].

5.2.4. Formally verified structural abstract interpretation

Participant: Yves Bertot.

We implemented a structurally recursive abstract interpreter for a simple language of while constructs, sequences and assignments. This work is a refinement of a previous abstract interpreter, more concise and with a more elegant proof. Lecture notes attached to this work are currently available as a research report and have been submitted for publication in the lecture notes of the Lernet summer school.

5.2.5. Properties of Gene networks

Participants: Yves Bertot, Maxime Dénès, Benjamin Lesage, Adrien Richard [Laboratoire CNRS I3S, Nice], Jean-Paul Comet [Université de Nice].

Genetic networks are an abstraction of the behavior of living organisms which provide a model of the conditions in which genes are expressed. Another abstract model relies on notions of finite-state automata. We studied the relations between the some aspects of the two models. The main result of this work is a Coq development where the proofs of two papers by Richard and Comet are formally verified. This work gives new insights on the formal modeling of genetic networks and will be continued. This work benefited from the support of the Color Program at INRIA Sophia Antipolis Méditerranée.

5.3. Tools for proof environments

5.3.1. Coqweb

Participant: Loïc Pottier.

Coqweb is a web interface to Coq, developed in collaboration with André Hirschowitz, Gang Xiao and Joachim Yameogo from the laboratory of mathematics of the University of Nice. It has been used since October 2004 by 10 teachers in mathematics at the University of Nice for 200 students in first year. The next version of this tool is visible at the following address:
http://pcmath170.unice.fr/wikimath

This year we developped and finished the mediawiki implementation of coqweb (mediawiki is the tool used by Wikipedia). We added records and inductive types to coqweb, so the whole Coq language can now be used in coqweb. Several contributors developed mathematics in this context, mainly at research level, but the activity with students is stopped due to a reorganisation of studies in first year at the university of Nice.

5.3.2. Gröbner bases

Participant: Loïc Pottier.

To prove automatically polynomial equalities in Coq via the nullstellensatz theorem of Hilbert, we connected Coq to three programs that computes Gröbner bases: F4, FGB, and gbcoq. The first two are C programs developed by J.C. Faugere. The third was partially extracted from the proof of Buchberger's algorithm by L.Théry. This lead to a Coq tactic, called gb, described in [11]. After that, we remarked that the computation of the whole Gröbner basis was not necessary. So we adapted gbcoq and, with the help of L.Théry and B.Grégoire, 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.

6. Other Grants and Activities

6.1. National initiatives

- We participate 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.
- We participate in the national contract CompCert, which started on Jan. 1st 2006. Other participants in this contract are the project-team CRISTAL (INRIA Rocquencourt), CEDRIC-CNAM (Evry), and PPS (Paris). The objective of this contract is to study the development of a formally verified compiler for a significant subset of C.
- We participate in the common laboratory between INRIA and Microsoft Research, in the Collaborative research action "Mathematical components". Other participants in this contract are the INRIA project-teams LOGICAL and PROVAL. The goals of this contract is to study the impact of small-scale reflective approaches to the formalisation of mathematics, especially in finite group theory and to experiment with extension of theorem provers with native arithmetics.
- 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.

6.2. European initiatives

Marelle participates in the network Types (type theory).

7. Dissemination

7.1. Conference and workshop attendance, travel

Yves Bertot, Laurence Rideau, Ioana Pasca, and Sidi Ould Biha attended the JFLA'08 conference (Journées Francophones des Langages Applicatifs) in Etretat, in January.

Nicolas Julien presented his work at the LORIA in Nancy, in January and at the LIP in Lyon, in February.

Yves Bertot gave a course at the Summer School Lernet, in Piriapolis (Uruguay), in February.

Vladimir Komendantsky attended the seminar on Logic and Interactions in Marseille, in February.

Ekaterina Komendantskay presented her work at the University of Edinburgh, in March.

Laurent Théry attended the TTVSI conference (Tools and Techniques for Verification of System Infrastructure), as invited speaker, in London, in March.

Vladimir Komendantsky and Ekaterina Komendantskaya attended the Types Workshop, in Torino, in March and presented their work.

Ekaterina Komendantskaya presented a paper at the CMCS workshop (Coalgebraic Methods in Computer Science), in Budapest, in April.

Yves Bertot gave a course at the University of Tsinghua (China), in April.

Nicolas Julien attended the FLOPS conference to present his work, in Nagoya, in April.

Yves Bertot gave a course at the School "Jeunes Chercheurs en Programmation", in Rennes, in May.

Ioana Pasca and Tuan Minh PHAM attended the School "Jeunes Chercheurs en Programmation", in Rennes, in May.

Vladimir Komendantsky attended the PPDP symposiun (Principles and Practice of Declarative Programming) and the PLID workshop (Programming Language Interference and Dependence) to present a paper, in Valencia, in July.

Vladimir Komendantsky was invited to give a talk at the University of Sevilla, in July.

Ekaterina Komendanskaya attended the NeSy workshop (Neural-Sumbolic Learning and Reasoning) at the ECAI conference to present her work, in Patras (Greece), in July.

Laurence Rideau, Laurent Théry, Yves Bertot, Ioana Pasca, and Maxime Denès visited the Microsoft Laboratory of Cambridge, in July.

Ioana Pasca gave a talk at the Mathlogaps training workshop, in Manchester, in July.

Yves Bertot, Sidi Ould Biha, Ioana Pasca, and Laurent Théry attented the TPHOLs conference (Theorem Proving in Higher Order Logics), in Montréal, in August, where they presented three papers.

7.2. Leadership within scientific community

- Yves Bertot was a member of the program committees for UITP'08, TPHOLs'08.
- Laurence Rideau was a member of the program committee for JFLA'08.
- Yves Bertot gave talks at the School for young researchers in programming on *introduction to type theory* and *Coq in a Hurry*.
- Yves Bertot was appointed as an external expert for the evaluation of an application as *docent* for Chalmers University, Sweden.
- Yves Bertot was appointed as an external expert for the evaluation of a grant for NWO, the Netherlands.
- Laurent Théry was appointed as an external expert for the evaluation of a grant by the Newton Institute (Cambridge, UK).
- Laurent Théry was appointed as an expert for the evaluation of an ANR grant, France.
- Project members reviewed papers for the journals JAR (Journal of Automated Reasoning), JSC (Journal of Symbolic Computation), TOMS (Transactions on Mathematical Software), for the lecture notes of the summer school Lernet, and for the conferences JFLA (Journées Francophones des Langages Applicatifs), TPHOLs (Theorem Proving in Higher Order Logics), ICFP (Internation Conference on Functional Programming), MSFP (Mathematical Structures of Functional Programming), Types, UITP (User-Interfaces for Theorem Provers).

7.3. Miscellaneous

- Yves Bertot acted as reviewer ("rapporteur") for the PhD thesis of Dorina Ghindici at the University of Lille.

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 2006, Oct. 1st, with funding from the French ministry of research and a teaching assistant grant.
- Yves Bertot supervises the Ph.D. project of Ioana Pasca, which started on Oct. 1st, with funding from the French ministry of research.

- Yves Bertot supervises the Ph.D project of Tuan Minh Pham, wich started on March 1st, with funding from ANR GALAPAGOS.

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 II* (Programming language semantics II), 2nd year Master (5th year, 24 hours), University of Nice.

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).

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 *Formal methods and advanced programming languages*, University of L'Aquila, Italy (18 hours), *Proof Mechanization*, 2nd year Master, University of Marseilles (3 hours). *Introduction to Coq*, École des Mines de Paris, (3 hours).

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

Articles in International Peer-Reviewed Journal

[4] L. RIDEAU, B. P. SERPETTE, X. LEROY. *Tilting at windmills with Coq: formal verification of a compilation algorithm for parallel moves*, in "Journal of Automated Reasoning", vol. 40, n^{o} 4, 2008, p. 307–326, http://gallium.inria.fr/~xleroy/publi/parallel-move.pdf.

Invited Conferences

[5] Y. BERTOT. *A Short Presentation of Coq*, in "Theorem Proving in Higher Order Logics, 21st International Conference, TPHOLs 2008, Montreal, Canada, August 18-21, 2008. Proceedings", Lecture Notes in Computer Science, vol. 5170, Springer, 2008, p. 12-16.

International Peer-Reviewed Conference/Proceedings

[6] Y. BERTOT, G. GONTHIER, S. O. BIHA, I. PAŞCA. *Canonical Big Operators*, in "Proceedings of the 21st International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2008)", Lecture Notes in Computer Science, vol. 5170, Springer, August 2008, p. 12–16, http://hal.inria.fr/inria-00331193/.

[7] Y. BERTOT, E. KOMENDANTSKAYA. *Inductive and Coinductive Components of Corecursive Functions in Coq*, in "Proceedings of CMCS'08, the 9th International Workshop on Coalgebraic Methods in Computer Science", ENTCS, vol. 203, April 2008, p. 25 - 47, http://hal.inria.fr/inria-00277075/en/.

[8] Y. BERTOT, V. KOMENDANTSKY. *Fixed point semantics and partial recursion in Coq*, in "PPDP '08: Proceedings of the 10th international ACM SIGPLAN conference on Principles and practice of declarative programming, New York, NY, USA", ACM, 2008, p. 89–96, http://hal.inria.fr/inria-00190975/.

[9] Y. BERTOT, L. THÉRY. *Dependent Types, Theorem Proving, and Applications for a Verifying Compiler*, in "1st IFIP TC 2/WG 2.3 Conference, on Verified Software: Theories, Tools, Experiments (VSTTE)", LNCS, vol. 4171, Springer, 2008, p. 173–181.

[10] N. JULIEN. *Certified exact real arithmetic using co-induction in arbitrary integer base*, in "Functional and Logic Programming Symposium (FLOPS)", Lecture Notes in Computer Science, Springer, 2008, http://hal.inria.fr/inria-00202744.

[11] L. POTTIER. *Connecting Gröbner Bases Programs with Coq to do Proofs in Algebra, Geometry and Arithmetics*, in "Proceedings of the LPAR Workshops: Knowledge Exchange: Automated Provers and Proof Assistants, and The 7th International Workshop on the Implementation of Logics", G. SUTCLIFFE, P. RUDNICKI, R. SCHMIDT, B. KONEV, S. SCHULZ (editors), CEUR Workshop Proceedings, n^{o} 418, 2008.

[12] L. THÉRY. *Proof Pearl. Revisiting the Mini-Rubik in Coq*, in "Proceedings of the 21st International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2008)", LNCS, vol. 5170, August 2008.