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

Head of project team
Yves Bertot [Research scientist INRIA, HdR]

Vice-head of project team
Laurence Rideau [Research scientist INRIA]

Administrative Assistant
Nathalie Bellesso [Administrative assistant]

Staff members INRIA
Loïc Pottier [Research scientist INRIA, HdR]
Laurent Théry [Research scientist INRIA]

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

Ph.D. students
Sidi Ould Biha [Phd. student, advised by L. Théry]
Nicolas Julien [Teaching Assistant, advised by Y. Bertot]
Ioana Pasca [Phd. student, arrived on Oct. 1st, advised by Y. Bertot]

Master students
Ioana Pasca [Master in Computer Science at the University of Nice, supervised by Y. Bertot]
Cody Roux [Master in Mathematics at the University of Nice, supervised by L. Pottier]
Abhishek Prateek [Bachelor in Computer Science, I.I.T. Delhi, supervised by Y. Bertot and L. Rideau]

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. Efficient Numbers in Coq

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

We have integrated our library for efficient computation inside Coq. This library includes representations for natural numbers, integers, and rationals based on binary trees. This library is now part of the Coq distribution.

5.1.2. Typing with ambiguities

Participants: Loïc Pottier, Cody Roux.

The main originality of coqweb is the language used to write mathematics, which is based on type theory with ambiguities. The algorithm for ambiguous typing which is integrated to coqweb has been further investigated. This year we concentrated on typing in presence of coercions. This problem is open even for simple types. We propose a novative rule system that makes it possible to infer types in many cases. An article will be published in the next JFLA conference [8].

5.1.3. Group theory

Participants: Georges Gonthier [Microsoft Research], Assia Mahboubi [INRIA MSR common laboratory], 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.

This year, we studied the formal proofs of the simplicity of the alternating groups A_n, for $n \geq 5$. This implied designing formal description for several new concepts: transitivity and $m-$transitivity of permutations, primitivity, imprimitivity, and collections of lemmas about these.

We formalized and proved Maschke's theorem, one of the fundamental theorems of the group representation theory. This required the development of libraries (formalizations and proof) of algebraic structures: Groups, Rings, Fields, Modules and Vector Spaces.

In all, we have completed a formal description of all the concepts and theorems found in an introductory course on group theory (group, subgroup, coset, Sylow group, permutation group). This work is described in [4].

5.1.4. Formalizing matrix algebra

Participants: Yves Bertot, Georges Gonthier [Microsoft Research], Sidi Ould Biha.

In the context of the "Mathematical Components" project, we studied the description of matrices as objects that can be alternatively viewed as finite graphs and as functions from a finite type to a type where equality is decidable. We then studied the description of determinants on this data-type as an iterated operation, summing values that are themselves obtained using iterated products. In this process, we also benefited of the previous formalization of the permutations of a finite type and their description as a finite type.

As a result, we were able to define a general notion of determinant and prove the main usual properties of determinants: multi-linearity, alternatedness, and the Cramer formula.

As the next step, we addressed the more significant theorem about characteristic polynomials known as the Cayley-Hamilton theorem [6]. We were able to formalize an elegant proof of this theorem by proving that the natural bijection between matrix with polynomial coefficients and polynomial with matrix coefficients is an isomorphism. This is the first formalization of this important theorem of algebra. This work on indexed operations and matrices also had a positive influence on our work in multi-variate analysis. We also expect this work to initiate our study of character theory.

5.1.5. 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. In this formal development we covered essential aspects of complete preordered structures, and monotonic and continuous functions. We provided a classical proof in Coq of the completeness of the flat complete preorder over an arbitrary type. In this development we structured and systematized the earlier work on the flat cpo done by Yves Bertot. In particular, the structure of our new development rests upon the implicit coercion mechanism of Coq. Thus we completed a part of the formal certification task and proved all foundational theorems concerning preorders and the fixed point functional. Also we found an ad hoc method of proving monotonicity and continuity of functionals, for which we provided a few illustrative examples. A description of this work has been submitted as a conference paper [10]. Also we studied the theory of fixed points in a generic category-theoretic setting, revisiting results due to P. Mulry and P. Freyd. We have found some ideas quite helpful for the purposes of our Coq development.

5.2. Verification of scientific algorithms

5.2.1. Co-recursion and real numbers

Participants: Yves Bertot, Nicolas Julien.

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. In particular, we concentrate on the implementation of series, which can then be used to define multiplication division, and other general purpose functions like exponentials, sine and cosine. The main work of this year was to define the inverse function and optimize the existing algorithms. The first stages of this work were also published in [5].

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. However, this method is not guaranteed to converge. The Kantorovitch theorem is one of the theorems that give sufficient conditions for the method to provide a good result and for this result to be unique.

We studied how this theorem could be described formally by first concentrating on the case of a function of one variable and then addressing the generalization to the multi-variate case. Most of the proof carries over easily from one variable to several variables, however some notions, which rely on matrices, their norms, and their invertibility, are more complex. This work benefited greatly from the experiments we do in algebra around matrices, determinants, and indexed 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. This work is published as [7].

5.2.3. Primality Proving with Elliptic Curves

Participants: Laurent Théry, Guillaume Hanrot [project-team CACAO].

We applied our formalisation of elliptic curves inside Coq to primality proving. Elliptic curves are currently the best way to get a certificate that a given number is prime. This is done using the Goldwasser-Kilian algorithm. It is now possible to verify these certificates inside Coq. Thanks to a relatively efficient computation mechanism, it is now possible to prove prime in Coq any prime number with less than 200 decimal digits. This works is described in [9].

5.2.4. Certified 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. A long article describing this work has been written and submitted for publication in a Journal.

5.2.5. An integrated study of semantic styles

Participant: Yves Bertot.

We improved the modularity of our collection of semantic descriptions of a small programming language, to show how they could be connected with types of strings, so that the proof extraction mechanism makes it possible to construct certified tools, although they rely on on a different type of strings.

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. This tool is visible at the following address:
http://pcmath170.unice.fr/wcw/spikini

New investigations on this experiment concern corrections and extension of ambiguous typing algorithm and the addition of structures that are similar to Coq's records. The proof text production has been re-done to exploit the advantages of the GF tool provided by A. Ranta from Chalmers University in Göteborg. About fifty pages of a mathematics course have been entered in this tool, with the exercises and their solutions (which are formal proofs). The level of these pages varies from the first year of the License program to the second year of the Master program.

We are currently porting the wiki encapsulation to Mediawiki, the same wiki framework as Wikipedia.
http://pcmath170.unice.fr/wikimath

This work was described in a paper presented at PATE'07 in Paris in June [3] and at the Types workshop in Edinburgh in October. We expect to participate to a European project proposal on this topic.

5.3.2. Connecting to computer algebra systems

Participants: Sidi Ould Biha, Laurent Théry.

GAP is a Computer Algebra System with particular emphasis on Computational Group Theory. We developed a library which aims to be the kernel of a broker between Coq and GAP. This library is written in C and uses XML and the new Coq command "external" to ensure communication between the two systems. As a first step the library can import in Coq a finite group structure generated by GAP. The finite group is represented by the matrix of composition of its elements by its internal law. Coq must simply check that this structure respects the group axioms. With the library, we were able to import into Coq groups of size 300, which require XML files of about 76 MB and a verifications of about 10 hours.

5.3.3. Connections between Coq and Eclipse

Participants: Yves Bertot, Laurence Rideau, Abhishek Prateek.

Eclipse is a popular programming environment, used mainly for Java, but also adaptable to other programming language. A team of developers based mainly in Edinburgh and Bremen proposed a package called PGIP (for *Proof General Interface Protocol* to integrate theorem provers in this programming environment, concentrating mainly on Isabelle. We studied how this package could be adapted to the Coq theorem prover. As a result, we obtained a small prototype that could be used to perform some of the basic tasks in formal proof development. This prototype demonstrated the feasibility of the approach but it also highlighted a few design problems with the current version of PGIP.

5.3.4. Sum of Squares

Participant: Laurent Théry.

A very effective way to prove the positivity of a polynomial $P(x) \geq 0$ is to write P as a sum of square. For example, to prove $x^2 + 3 * x + 2 >= 0$ we use the fact $x^2 + 3 * x + 2 = 1/8((3 * x + 4 * 1)^2 + 23 * x^2)$. We have integrated the library initially developed by John Harrison for HOL-LIGHT inside COQ.

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 formalization 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 attended the commemorative workshop in honor of Gilles Kahn in Paris in February, the Types conference in Udine, Italy, in May, the MEMOCODE conference in Nice in June, the TPHOLs conference in Kaiserslautern, Germany, in September.

Nicolas Julien attended the JFLA'07 conference (Journées Francophones des Langages applicatifs) in Aix-les-Bains and the RAIM workshop (Rencontres sur l'Arithmétique de l'informatique mathématique) in Montpellier in January, the Types'07 conference in May, the spring school in Programming ("Ecole jeunes chercheurs en Programmation") in Dinard and Rennes in June, and the Types summer school in Bertinoro, Italy, in August.

Sidi Ould Biha attended the Types'07 conference in May, the spring school in Programming in June, and the Types summer school in August.

Ioana Pasca attended the "CEA-EDF-INRIA" school on certified numerical computation in Nancy in October.

Loïc Pottier attended the PATE conference in Paris in June and the Types Workshop "Mathwiki" in Edinburgh in October.

Laurence Rideau attended the workshop in honor of Gilles Kahn in January, the Types conference in May, and the TPHOLs conference in September.

Laurent Théry attended the workshop in honor of Gilles Kahn in January and the TPHOLs conference in September.

7.2. Leadership within scientific community

- Yves Bertot was a member of the program committees for VERIFY'07, TPHOLs'07.
- 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 invited for a one week intensive course at the University of Chalmers, as part of the TYPES European cooperation action, on the formal description of programming language semantics.
- Yves Bertot was appointed as steering committee member for the UITP series of conferences.
- Laurent Théry gave an invited talk at the Mathlogaps training workshop in Lyon in June.
- Laurent Théry gave an invited talk at École Normale Supérieure in Lyon in October.
- Laurent Théry was appointed as steering committee member for the TPHOLs series of conferences.
- Project members reviewed papers for the journals JFP (Journal of Functional Programming), JSC (Journal of Symbolic Computation), LMCS (Logical Methods in Computer Science), SCP (Science of Computer Programming), TOMS (Transactions on Mathematical Software), for a book in memory in Gilles Kahn (in preparation), and for the conferences ASCM (Asian Symposium on Computer Mathematics), CHES (Cryptographic Hardware and Embedded Systems), JFLA (Journées Francophones des Langages Applicatifs), STACS'08 (Symposium on Theoretical Aspects of Computer Science), TPHOLs (Theorem Proving in Higher Order Logics), Types, VERIFY.
- Yves Bertot is a co-editor for a book in memory of Gilles Kahn, to appear in early 2008.

7.3. Miscellaneous

- Yves Bertot was a member of the *Conseil National des Universités* (National University Council), 27th section. This position brings more than a month of work in reviewing nationwide applications for university professor positions.
- Yves Bertot acted as reviewer (“rapporteur”) for the PhD theses of Houda Anoun at Bordeaux university and Delphine Longuet at Evry university.

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.

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, *Proof mechanization*, 2nd year Master (5th year, 3 hours) University of Marseille, *Introduction to Coq* École des Mines (3 hours).

Nicolas Julien *Object oriented programming*, *Introduction to programming in C*, *Computer architecture*.

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 (40 hours), *Proof Mechanization*, 2nd year Master, University of Marseilles (3 hours). *Introduction to Coq*, École des Mines, (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] L. THÉRY. *A Machine-Checked Implementation of Buchberger’s Algorithm*, in "Journal of Automated Reasoning", vol. 26, 2001, p. 107–137.

Year Publications

Publications in Conferences and Workshops

[3] J. BLANC, J. GIACOMETTI, A. HIRSCHOWITZ, L. POTTIER. *Proofs for freshmen with Coqweb*, in "Proceedings of the International Workshop on Proof Assistants and Types in Education, June 25th, 2007, associated workshop of the 2007 Federated Conference on Rewriting, Deduction and Programming, CNAM Paris, France", H. GEUVERS, P. COURTIEU (editors), June 2007, http://www.cs.ru.nl/~herman/PUBS/proceedingsPATE.pdf.

[4] 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.

[5] N. JULIEN. *Arithmétique réelle exacte certifiée, co-induction et base arbitraire*, in "Journées francophones pour les langages applicatifs, JFLA'07", 2007, http://hal.inria.fr/inria-00116820.

[6] S. OULD BIHA. *Formalisation des mathématiques : une preuve du théorème de Cayley-Hamilton*, in "Journées Francophones des Langages Applicatifs", to appear, January 2008.

[7] I. PAŞCA. *A Formal Verification for Kantorovitch's Theorem*, in "Journées Francophones des Langages Applicatifs", to appear, January 2008.

[8] C. ROUX. *Types, Logique et Coercions Implicites*, in "Journées Francophones des Langages Applicatifs", to appear, January 2008.

[9] L. THÉRY, G. HANROT. *Primality Proving with Elliptic Curves*, 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. 319-333, http://hal.inria.fr/inria-00138382.

Internal Reports

[10] Y. BERTOT, V. KOMENDANTSKY. *Fixed point semantics and partial recursion in Coq*, Draft paper, submitted to a conference, Technical report, INRIA, November 2007, http://hal.inria.fr/inria-00190975.

[11] L. RIDEAU, B. P. SERPETTE, X. LEROY. *Battling windmills with Coq: formal verification of a compilation algorithm for parallel moves*, Draft paper, submitted to a journal, Technical report, INRIA, September 2007, http://hal.inria.fr/inria-00176007/fr/.