Activity Report 2011

Project-Team MARELLE

Mathematical, Reasoning and Software

RESEARCH CENTER
Sophia Antipolis - Méditerranée

THEME
Programs, Verification and Proofs

Table of contents

Project-Team MARELLE

Keywords: Interactive Theorem Proving, Formal Methods, Security, Cryptography

1. Members

Research Scientists

Yves Bertot [Team leader, INRIA, HdR]
Benjamin Grégoire [Research scientist INRIA]
José Grimm [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*]

Technical Staff

Anne Pacalet [until November 2011]

PhD Students

Guillaume Cano [supervised by Y. Bertot]
Maxime Dénès [supervised by Y. Bertot]
Nicolas Julien [supervised by Y. Bertot]
Sylvain Heraud [supervised by B. Grégoire et Y. Bertot, until December 2011]
Tuan Minh Pham [supervised by Y. Bertot,until December 2011]
Jorge Luis Sacchini [supervised by B. Grégoire]
Michaël Armand [supervised by L. Théry and B. Grégoire]

Administrative Assistant

Nathalie Bellesso [Administrative assistant]

2. Overall Objectives

2.1. Highlights

Our work on formal proofs for cryptography now receives attention in best conferences of specialists of that domain.
BEST PAPER AWARD :
[11] **Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings**. G. BARTHE, B. GRÉGOIRE, S. HERAUD, S. ZANELLA BÉGUELIN.

2.2. Introduction

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 mechanically 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 a 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 investigate the algorithms that are present in programming language implementations, 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 accuracy 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. Software

5.1. Semantics

Participant: Yves Bertot [correspondant].

This is a library for the Coq system, where the description of a toy programming language is presented. The value of this library is that it can be re-used in classrooms to teach programming language semantics or the Coq system. The topics covered include introductory notions to domain theory, pre and post-conditions, abstract interpretation, and the proofs of consistency between all these point of views on the same programming language. Standalone tools for the object programming language can be derived from this development.
See also the web page http://coq.inria.fr/pylons/pylons/contribs/view/Semantics/v8.3.

- ACM: F3.2 F4.1
- AMS: 68N30
- Programming language: Coq

5.2. Certicrypt

Participants: Gilles Barthe [IMDEA Software institute], Juan Manuel Crespo [IMDEA Software institute], Benjamin Grégoire [correspondant], Sylvain Heraud, César Kunz [IMDEA Software institute], Federico Olmedo [IMDEA Software institute], Santiago Zanella Béguelin [IMDEA Software institute].

CertiCrypt takes a language-based approach to cryptography: the security of a cryptographic scheme and the cryptographic assumptions upon which its security relies are expressed by means of probabilistic programs, called games; in a similar way, adversarial models are specified in terms of complexity classes, e.g. probabilistic polynomial-time programs. This code-centric view leads to statements that are amenable to formalization and tool-assisted verification. CertiCrypt instruments a rich set of verification techniques for probabilistic programs, including equational theories of observational equivalence, relational Hoare logic, data-flow analysis-based program transformations, and game-based techniques such as eager/lazy sampling and failure events.

See also the web page http://easycrypt.gforge.inria.fr/.

6. New Results

6.1. Type theory and formalization of mathematics

6.1.1. Foundational aspects of mechanized proofs

Participants: José Grimm, Loïc Pottier.

We attempt to prove all theorems in the “Theory of Sets” of Bourbaki. The first chapter descripes Formal Mathematics, and we show that it can be interpreted in the Coq language, thanks to a bunch of axioms introduced by Carlos Simpson (CNRS, Nice), modulo some modifications. This work that was started in 2009, when J. Grimm was in the Apics project-team. A new formulation of this work using `ssreflect` has proved more efficient than the initial formulation relying on standard Coq.

The second chapter of Bourbaki covers the theory of sets, *per se*. It defines ordered pairs, correspondences, unions, intersections and products of a family of sets, as well as equivalence relations. The work of formalizing this chapter comprises 15000 lines of Coq script and is described in a technical report and a paper for the journal of formal reasoning published in 2010.

The third chapter of Bourbaki covers the theory of ordered sets, well-ordered sets, equipotent sets, cardinals, natural integers, and infinite sets; its implementation in Coq is described in [21]. This chapter is longer (22000 lines of code), and there are more exercises (18000 lines of code for about half of the exercises currently implemented).

We also looked at the *univalent foundation* proposed by V. Voevodsky to provide a new model for equality in type theory and simplified the proof that he proposed to derive extensionality from the univalence axiom.

6.1.2. Group theory (Character theory)

Participants: Georges Gonthier [Microsoft Research], Laurence Rideau, Laurent Théry.

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 [4].

This year, we have initiated the formalisation of the second book of the proof of Feit-Thompson’s theorem. The basic properties of character theories are now covered. This lets us formalised the first 4 chapters of the second book, “Character theory for the Odd Order Theorem” by Peterfalvi.

6.1.3. Proofs in geometry

Participants: Tuan Minh Pham, Yves Bertot.

The work on elementary (synthetic) geometry has been completed. A publication on the topic has also been presented at a conference [19]. This work was also the main content of Tuan Minh Pham’s thesis which was defended in November [5].

6.1.4. Towards constructive algebraic topology

Participants: Laurence Rideau, Maxime Dénès, Yves Bertot.

We have participated in the formalization of a complete chain of computation from an image (as a bitmap) to the corresponding Betti numbers and homology groups. In particular, we improved the formalization of “incidence simplicial matrices” in `ssreflect`. This work was described in conference article [17].

6.1.5. Computing with polynomials and matrices

Participants: Maxime Dénès, Yves Bertot.

The libraries of the project "Mathematical Components" propose a rather complete formalisation of polynomials and matrices. Unfortunately, these objects cannot be used directly for computing.

We have continued our study of executable algorithms to compute with matrices and polynomials inside Coq. In collaboration with other members of the European project Formath, we have looked at implementation of Strassen-Winograd and Karatsuba for fast matrix multiplication and other algorithms for various kinds of matrix normal forms: Smith normal form, Frobenius, and Jordan normal forms. This work is described in an article that has been submitted for publication.

6.1.6. Regularity of interval matrices

Participants: Guillaume Cano, Yves Bertot.

As part of our work on the regularity of interval matrices, we still needed to formalize the Perron-Frobenius theorem. This year we concentrated on an important lemma for this formalization, the Bolzano-Weierstrass theorem, which requires a usable formalization of general topology, in particular the concept of compact.

6.1.7. Type-based termination

Participants: Jorge Luis Sacchini, Benjamin Grégoire.

The work on this topic has been completed and is described in Jorge-Luis Sacchini’s Ph.D thesis, which was defended in June 2011 [6].

6.1.8. Native compilation of terms with primitive structures

Participants: Mathieu Boespflug [McGill University, Canada], Maxime Dénès, Benjamin Grégoire.

We kept working on the integration of the native compiler of the Ocaml language into a scheme for the efficient reduction of terms in the calculus of inductive constructions. This work is described in a publication at the conference CPP11 in Taiwan [14].

6.2. Proving tools

6.2.1. Connecting an SMT prover and Coq

Participants: Michaël Armand, Germain Faure [project-team Typical], Benjamin Grégoire, Chantal Keller [project-team Typical], Laurent Théry.

Our previous work on integrating SAT technology has been used as a basis to obtain SMT automation within Coq. We are now capable of replaying traces produced by the SMT prover VERIT that deal with conjunctive normal forms, congruence closures, and linear arithmetic. We are actively working on adding quantified formulae. This work is supported by the ANR Decert project. A prelimary version [10] of this work has been presented at the workshop PSATTT’11, a full version [9] at the conference CPP11. The generic exchange proof format [13] for SMT has been presented at the workshop PXTP’11.

6.2.2. Geometric Algebras and Automatic Theorem Proving

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

We have completed our work on Grassman-Cayley algebras. This has been published in the post-proceedings of the ADG’10 conference. We are now working on the natural continuation of this work: Clifford’s algebras. We have very encouraging premilary results.

6.2.3. Taylor models in Coq

Participants: Erik Martin-Dorel [project-team Arénaire], Ioana Paşca [project-team Arénaire], Micaela Mayero [Université Paris XIII], Laurence Rideau, Laurent Théry.

Taylor models are a very effective way to approximate real functions with polynomials. We have started a formalisation of these models in the Coq prover. In a first step, we have concentrated our efforts in having a computational version of these models within Coq using native computations, certified floating point and interval arithmetics. Since our first evaluations show that they behave well computationally, we are now working on completing this work with the corresponding correctness proofs. This work is supported by the ANR Tamadi.

6.2.4. Tactics on polynomial equalities: nsatz

Participant: Loïc Pottier.

We started describing in the Coq programming language an efficient algorithm to compute Gröbner bases, similar to the one written in ocaml for the nsatz tactic. We hope to prove it correct and to use it for proofs by reflexion in commutative algebra.

6.2.5. D-Modules

Participant: Loïc Pottier.

We studied normalization of non-commutative polynomials ad exponentials in the Weyl algebra. The normal forms we found are similar with the one described found by Blasiak and Flajolet for graph models.

6.3. Formal study of cryptography

6.3.1. Certicrypt

Participants: Gilles Barthe, Benjamin Grégoire, Sylvain Heraud, Santiago Zanella.

CertiCrypt is a general framework to certify the security of cryptographic primitives in the Coq proof assistant.

We completed a machine-checked proof of the security of OAEP (a widely public-key encryption scheme based on trapdoor permutations) against adaptive chosen ciphertext attacks under the assumption that the underlying permutation is partial-domain one-way. This work has been described in a publication at the conference CT-RSA 2011 in San Francisco [12].

6.3.1.1. Easycrypt

Participants: Gilles Barthe [IMDEA], Benjamin Grégoire, Sylvain Heraud, Anne Pacalet, Santiago Zanella.

Based on our experience with Certicrypt, we started last year the development of the tool Easycrypt. The goal of this work is to provide a friendly tool easily usable by cryptographers without knowledge of formal proof assistants. The idea is to use the techniques formally proved in Certycrypt and to call SMT-provers instead of using Coq. We have applied Easycrypt on a variety of academic examples and one bigger example: the proof of IND-CCA security of the Cramer-Shoup cryptosystem. The drawback of this tool is that it provide less guarantees than Certicrypt for the correctness of the proof. To fill this gap we are now able to generate Coq files (based on Certicrypt) allowing to check the validity of Easycrypt proofs. This work has been described in a publication at the conference CRYPTO 2011 in Santa Barbara and has obtained the best paper Award [11].

7. Partnerships and Cooperations

7.1. National Initiatives

- We were the leader of the ANR project Galapagos, which started on Nov. 19th 2007 and finished on Nov. 19th 2011. Other participants in this contract are the universities of Strasbourg and Poitiers, 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 participated to the ANR SCALP, which started on January 1st, 2008. Other participants in this contract were DCS-Verimag (Grenoble), Plume-LIP (Lyon), Proval-LRI (Orsay), CPR-Cédric (Cnam, Paris). In this project we focused on the formalization of Cryptography.
- We participated to the ANR project DeCert, which started on January 2009. Other participants are CEA List (Paris), LORIA-INRIA (Nancy), Celtique (IRISA Rennes), Proval (LRI Orsay), Typical (INRIA Saclay), Systerel (Aix-en-provence). The objective of the DeCert project was 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 participate to the ANR project TAMADI, which started in October 2010. Other participants are ARENAIRE-INRIA Rhone-Alpes and the PEQUAN team from University of Paris VI Pierre and Marie Curie. The objective of the TAMADI project is to study question of precision in floating-point arithmetic and to provide formal proofs on this topic.

7.2. European Initiatives

7.2.1. FP7 Projet

7.2.1.1. FORMATH

Title: Formath

Type: COOPERATION (ICT)

Defi: FET Open

Instrument: Specific Targeted Research Project (STREP)

Duration: March 2010 - February 2013

Coordinator: University of Götegorg (Sweden)

Others partners: Radboud University Nijmegen, (the Netherlands), University of La Rioja, (Spain).

See also: http://wiki.portal.chalmers.se/cse/pmwiki.php/ForMath/ForMath

Abstract: The objective of this project is to develop libraries of formalised mathematics concerning algebra, linear algebra, real number computation, and algebraic topology. The libraries that we plan to develop in this proposal are especially chosen to have long-term applications in areas where software interacts with the physical world. The main originality of the work is to structure these libraries as a software development, relying on a basis that has already shown its power in the formal proof of the four-colour theorem, and to address topics that were mostly left untouched by previous research in formal proof or formal methods.

7.2.2. Major European Organizations with which you have followed Collaborations

Chalmers University, Programming Logic Group, (Sweden)

Type Theory and its application to formalizing of mathematics, especially algebraic concepts.

Radboud University, ICIS, Foundations group, (the Netherlands)

Type theory and its application to formalizing mathematics, especially numeric computation.

University of La Rioja, Programming and Symbolic Computation Team, (Spain)

Formal study of algebraic algorithms and application to algebraic topology.

8. Dissemination

8.1. Animation of the scientific community

- Members of the project participated to the programm committees of Coq'11, ITP'11 (Interactive Theorem Proving), Thedu (Computer Theorem Proving Components for Educational Software), PxTP (Proof Exchange for Theorem Proving).
- Members of the project reviewed papers for the conferences SAC'11 (Symposium on Applied Computing), ISSAC'11 (International Symposium on Symbolic and Algebraic Computation), ITP'11.
- Members of the project reviewed papers for the journal JUCS (Journal of Universal Computer Science),
- Members of the project wrote reports for the "habilitation à diriger des recherches" of J.-C. Filliatre. They were members of the Jury for the PhD defense of Arnaud Spiwack, Muhammad Khan.
- Yves Bertot and Benjamin Grégoire taught at the CEA-EDF-Inria School on "Modelling and Verifying Algorithms in Coq: an introduction" in November in Paris, France.
- Yves Bertot taught at the 3rd Asian-Pacific Summer School on Formal Methods, in Suzhou, China.
- Members of the project attended the conferences JFLA, Coq-3, ITP, CADE, Types, CRYPTO, CPP (Certified Programs and Proofs), CICM (Conference on Intelligent Computer Mathematics), ICCSA (International Conference on Computational Science and its Applications).
- Maxime Dénès gave an invited talk at Ecole Normale Supérieure in Lyons, France.

8.2. Teaching

Master (or equivalent) : "Logic", 70 hours (Loïc Pottier) M2, University of Nice Sophia Antipolis, France

Master (or equivalent) : "Programming language semantics", 30 hours (Yves Bertot), 12 hours (Maxime Dénès) M1, University of Nice Sophia Antipolis, France

Doctorate (or equivalent) : CEA-EDF-Inria "Modeling and Verifying Algorithms in Coq: an introduction", 29 heures (Benjamin Grégoire), 12 heures (Yves Bertot), Inria, France

PhD : Jorge-Luis Sacchini, "On Type-Based Termination and Dependent Pattern Matching in the Calculus of Inductive Constructions", ParisTech (Mines), defended on June 29th, 2011, supervised by G. Barthe (IMDEA), and BenjaminGrégoire.

PhD : Tuan MinhPham, "Description formelle de propriétés géométriques", University of Nice Sophia Antipolis, defended on November 21st, 2011, supervised by YvesBertot

PhD in progress : Michaël Armand, "Application de la certification de résultats à l'automatisation de preuves interactives" , started in in October 2009, supervised by Laurent Théry and Benjamin Grégoire.

PhD in progress : Sylvain Heraud, "Vérification semi-automatique de primitives cryptographiques", started in October 2008, supervised by Benjamin Grégoire and Yves Bertot.

PhD in progress : Guillaume Cano, “Une formalisation adaptable de l’algèbre linéaire”, started in October 2010, supervised by Yves Bertot.

PhD in progress : Maxime Dénès, “Description formelle d’algèbre linéaire pour l’algorithmique rapide”, started in September 2010, supervised by Yves Bertot.

9. Bibliography

Major publications by the team in recent years

[1] 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.

[2] Y. Bertot, P. Castéran. *Interactive Theorem Proving and Program Development, Coq’Art:the Calculus of Inductive Constructions*, Springer-Verlag, 2004.

[3] 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, Springer, August 2008, vol. 5170, p. 12–16, http://hal.inria.fr/inria-00331193/.

[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, Springer-Verlag, September 2007, vol. 4732, p. 86-101, http://hal.inria.fr/inria-00139131.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[5] T. M. Pham. *Description formelle de propriétés géométriques*, Université de Nice Sophia Antipolis, September 2011.

[6] J. Sacchini. *Terminaison basée sur les types et filtrage dépendant pour le calcul des constructions inductives*, École Nationale Supérieure des Mines de Paris, June 2011, http://hal.inria.fr/pastel-00622429/en.

Articles in International Peer-Reviewed Journal

[7] Y. Bertot, F. Guilhot, A. Mahboubi. *A formal study of Bernstein coefficients and polynomials*, in "Mathematical Structures in Computer Science", 2011, vol. 21, n^{o} 04, p. 731-761, http://hal.inria.fr/inria-00503017/en.

[8] J. O. Blech, B. Grégoire. *Certifying compilers using higher-order theorem provers as certificate checkers*, in "Formal Methods in System Design", 2011, vol. 38, n^{o} 1, p. 33-61.

International Conferences with Proceedings

[9] M. Armand, G. Faure, B. Grégoire, C. Keller, L. Théry, B. Werner. *A Modular Integration of SAT/SMT Solvers to Coq through Proof Witnesses*, in "First International Conference on Certified Programs and Proofs", Tawain, Lecture Notes in Computer Science, Springer, December 7-9 2011, To Appear.

[10] M. Armand, G. Faure, B. Grégoire, C. Keller, L. Théry, B. Werner. *Verifying SAT and SMT in Coq for a fully automated decision procedure*, in "PSATTT'11: International Workshop on Proof-Search in Axiomatic Theories and Type Theories", Wroclaw, Poland, Germain Faure, Stéphane Lengrand, Assia Mahboubi, 2011, http://hal.inria.fr/inria-00614041/en.

[11] *Best Paper*
G. Barthe, B. Grégoire, S. Heraud, S. Zanella Béguelin. *Computer-Aided Security Proofs for the Working Cryptographer*, in "Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings", Lecture Notes in Computer Science, Springer, 2011, vol. 6841, p. 71-90, Best Paper Award.

[12] G. Barthe, B. Grégoire, Y. Lakhnech, S. Zanella Béguelin. *Beyond Provable Security Verifiable IND-CCA Security of OAEP*, in "Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011", San Francisco, CA, USA, Springer, February 14-18 2011, vol. 6558, p. 180-196.

[13] F. Besson, P. Fontaine, L. Théry. *A Flexible Proof Format for SMT: a Proposal*, in "PxTP 2011: First InternationalWorkshop on Proof eXchange for Theorem Proving", Wroclaw, Poland, 2011, http://pxtp2011.loria.fr/PxTP2011.pdf.

[14] M. Boespflug, M. Dénès, B. Grégoire. *Full reduction at full throttle*, in "First International Conference on Certified Programs and Proofs", Tawain, Lecture Notes in Computer Science, Springer, December 7-9 2011, To Appear.

[15] L. Fuchs, L. Théry. *A Formalization of Grassmann-Cayley Algebra in COQ and Its Application to Theorem Proving in Projective Geometry*, in "Automated Deduction in Geometry", Lecture Notes in Computer Science, 2011, vol. 6877, p. 51-67.

[16] B. Grégoire, L. Pottier, L. Théry. *Proof Certificates for Algebra and Their Application to Automatic Geometry Theorem Proving*, in "Automated Deduction in Geometry - 7th International Workshop, ADG 2008", Shanghai, China, Lecture Notes in Computer Science, Springer, September 22-24 2011, vol. 6301, p. 42-59, Revised Papers.

[17] J. Heras, M. Poza, M. Dénès, L. Rideau. *Incidence simplicial matrices formalized in Coq/SSReflect*, in "Conference on Intelligent Computer Mathematics'11", Lecture Notes in Artificial Intelligence, Springer-Verlag, August 2011, vol. 6824, http://hal.inria.fr/inria-00603208/en.

[18] S. Heraud, D. Nowak. *A Formalization of Polytime Functions*, in "Interactive Theorem Proving - Second International Conference, ITP 2011", Berg en Dal, The Netherlands, M. C. J. D. van Eekelen, H. Geuvers, J. Schmaltz, F. Wiedijk (editors), Lecture Notes in Computer Science, August 22-25 2011, vol. 6898, p. 119-134.

[19] T. M. Pham, Y. Bertot, J. Narboux. *A Coq-based Library for Interactive and Automated Theorem Proving in Plane Geometry*, in "The 11th International Conference on Computational Science and Its Applications (ICCSA 2011)", Santander, Spain, Lecture Notes in Computer Science, Springer-Verlag, 2011, vol. 6785, p. 368-383 [*DOI :* 10.1007/978-3-642-21898-9_32], http://hal.inria.fr/inria-00584918/en.

Research Reports

[20] J. GRIMM. *Implementation of Bourbaki's Elements of Mathematics in Coq: Part One, Theory of Sets*, INRIA, 2011, n^{o} RR-6999, Version 5, http://hal.inria.fr/inria-00408143/en/.

[21] J. GRIMM. *Implementation of Bourbaki's Elements of Mathematics in Coq: Part Two; Ordered Sets, Cardinals, Integers*, INRIA, 2011, n^{o} RR-7150, version 4, http://hal.inria.fr/inria-00440786/en/.