Activity Report 2016

Team DEDUCTEAM

Deduction modulo, interopérabilité et démonstration automatique

Inria teams are typically groups of researchers working on the definition of a common project, and objectives, with the goal to arrive at the creation of a project-team. Such project-teams may include other partners (universities or research institutions).

RESEARCH CENTER
Saclay - Île-de-France

THEME
Proofs and Verification

Table of contents

Team DEDUCTEAM

Creation of the Team: 2011 December 01, updated into Project-Team: 2017 January 01

Keywords:

Computer Science and Digital Science:

2. - Software
2.1.2. - Object-oriented programming
2.1.3. - Functional programming
2.1.11. - Proof languages
2.4.3. - Proofs
3.1.1. - Modeling, representation
7. - Fundamental Algorithmics
7.4. - Logic in Computer Science
7.13. - Quantum algorithms

Other Research Topics and Application Domains:

7. - Transport and logistics

1. Members

Research Scientists

Gilles Dowek [Team Leader, Inria, Senior Researcher, HDR]
Frédéric Blanqui [Inria, Researcher, HDR]

PhD Students

Guillaume Bury [Univ. Paris VII]
Frédéric Gilbert [Min. Ecologie]
François Thiré [ENS Cachan, since October 2016]
Raphaël Cauderlier [CNAM, until August 2016]
Pierre Halmagrand [CNAM, until November 2016]

Post-Doctoral Fellow

Simon Martiel [Univ. Paris-Est Créteil]

Visiting Scientists

Jean-Pierre Jouannaud [Professor Emeritus, Univ. Paris-Saclay, HDR]
Guillaume Burel [ENSIIE, Associate Professor]
Catherine Dubois [ENSIIE, Professor, HDR]
Olivier Hermant [Mines ParisTech, Researcher]

Administrative Assistant

Thida Iem [Inria]

2. Overall Objectives

2.1. Objectives

The team investigates applications of recent results in proof theory to the design of logical frameworks and automated theorem proving systems. It develops the Dedukti logical framework and the iProver modulo and Zenon modulo automated theorem proving systems.

2.2. History

Deduction modulo is a formulation of predicate logic where deduction is performed modulo an equivalence relation defined on propositions. A typical example is the equivalence relation relating propositions differing only by a re-arrangement of brackets around additions, relating, for instance, the propositions $P((x + y) + z)$ and $P(x + (y + z))$. Reasoning modulo this equivalence relation permits to drop the associativity axiom. Thus, in Deduction modulo, a theory is formed with a set of axioms and an equivalence relation. When the set of axioms is empty the theory is called *purely computational*.

Deduction modulo was proposed at the end of the 20th century as a tool to simplify the completeness proof of equational resolution. Soon, it was noticed that this idea was also present in other areas of logic, such as Martin-Löf's type theory, where the equivalence relation is definitional equality, Prawitz' extended natural deduction, etc. More generally, Deduction modulo gives an account on the way reasoning and computation are articulated in a formal proof, a topic slightly neglected by logic, but of prime importance when proofs are computerized.

The early research on Deduction modulo focused on the design of general proof search methods—Resolution modulo, tableaux modulo, etc.—that could be applied to any theory formulated in Deduction modulo, to general proof normalization and cut elimination results, to the definitions of models taking the difference between reasoning and computation into account, and to the definition of specific theories—simple type theory, arithmetic, some versions of set theory, etc.—as purely computational theories.

3. Research Program

3.1. From proof-checking to Interoperability

A new turn with Deduction modulo was taken when the idea of reasoning modulo an arbitrary equivalence relation was applied to typed λ-calculi with dependent types, that permits to express proofs as algorithms, using the Brouwer-Heyting-Kolmogorov interpretation and the Curry-de Bruijn-Howard correspondence [27]. It was shown in 2007, that extending the simplest λ-calculus with dependent types, the $\lambda\Pi$-calculus, with an equivalence relation (more precisely a coingruence), led to a calculus we called the $\lambda\Pi$-calculus modulo, that permitted to simulate many other λ-calculi, such as the Calculus of Constructions, designed to express proofs in specific theories.

This led to the development of a general proof-checker based on the $\lambda\Pi$-calculus modulo [3], that could be used to verify proofs coming from different proof systems, such as Coq [26], HOL [33], etc. To emphasize this versatility of our proof-system, we called it Dedukti —"to deduce" in Esperanto. This system is currently developed together with companion systems, Coqine, Krajono, Holide, Focalide, and Zenonide, that permits to translate proofs from Coq, HOL, Focalize, and Zenon, to Dedukti. Other tools, such as Zenon Modulo, directly output proofs that can be checked by Dedukti. Dedukti proofs can also be exported to other systems, in particular to the MMT format [37].

A thesis, which is at the root of our research effort, and which was already formulated in [32] is that proof-checkers should be theory independent. This is for instance expressed in the title of our invited talk at Icalp 2012: *A theory independent Curry-De Bruijn-Howard correspondence*. Such a theory independent proof-checker is called a *Logical Framework*.

Using a single prover to check proofs coming from different provers naturally led to investigate how these proofs could interact one with another. This issue is of prime importance because developments in proof systems are getting bigger and, unlike other communities in computer science, the proof-checking community has given little effort in the direction of standardization and interoperability. On a longer term we believe that, for each proof, we should be able to identify the systems in which it can be expressed.

3.2. Automated theorem proving

Deduction modulo has originally been proposed to solve a problem in automated theorem proving and some of the early work in this area focused on the design of an automated theorem proving method called *Resolution modulo*, but this method was so complex that it was never implemented. This method was simplified in 2010 [5] and it could then be implemented. This implementation that builds on the iProver effort [36] is called iProver modulo.

iProver modulo gave surprisingly good results [4], so that we use it now to search for proofs in many areas: in the theory of classes—also known as B set theory—, on finite structures, etc. Similar ideas have also been implemented for the tableau method with in particular several extensions of the Zenon automated theorem prover. More precisely, two extensions have been realized: the first one is called SuperZenon [35] [30] and is an extension to superdeduction (which is a variant of Deduction modulo), and the second one is called ZenonModulo [28], [29] and is an extension to Deduction modulo. Both extensions have been extensively tested over first-order problems (of the TPTP library), and also provide good results in terms of number of proved problems. In particular, these tools provide good performances in set theory, so that SuperZenon has been successfully applied to verify B proof rules of Atelier B (work in collaboration with Siemens). Similarly, we plan to apply ZenonModulo in the framework of the BWare project to verify B proof obligations coming from the modeling of industrial applications.

More generally, we believe that proof-checking and automated theorem proving have a lot to learn from each other, because a proof is both a static linguistic object justifying the truth of a proposition and a dynamic process of proving this proposition.

3.3. Models of computation

The idea of Deduction modulo is that computation plays a major role in the foundations of mathematics. This led us to investigate the role played by computation in other sciences, in particular in physics. Some of this work can be seen as a continuation of Gandy's [31] on the fact that the physical Church-Turing thesis is a consequence of three principles of physics, two well-known: the homogeneity of space and time, and the existence of a bound on the velocity of information, and one more speculative: the existence of a bound on the density of information.

This led us to develop physically oriented models of computations.

4. Application Domains

4.1. Safety of aerospace systems

In parallel with this effort in logic and in the development of proof checkers and automated theorem proving systems, we always have been interested in using such tools. One of our favorite application domain is the safety of aerospace systems. Together with César Muñoz' team in Nasa-Langley, we have proved the correctness of several geometric algorithms used in air traffic control.

This has led us sometimes to develop such algorithms ourselves, and sometimes to develop tools for automating these proofs.

4.2. B-set theory

Set theory appears to be an appropriate theory for automated theorem provers based on Deduction modulo, in particular the several extensions of Zenon (SuperZenon and ZenonModulo). Modeling techniques using set theory are therefore good candidates to assess these tools. This is what we have done with the B method whose formalism relies on set theory. A collaboration with Siemens has been developed to automatically verify the B proof rules of Atelier B [34]. From this work presented in the Doctoral dissertation of Mélanie Jacquel, the SuperZenon tool [35] [30] has been designed in order to be able to reason modulo the B set theory. As a sequel of this work, we contribute to the BWare project whose aim is to provide a mechanized framework to support the automated verification of B proof obligations coming from the development of industrial applications. In this context, we have recently designed ZenonModulo [28], [29] (Pierre Halmagrand's PhD thesis, which has started on October 2013) to deal with the B set theory. In this work, the idea is to manually transform the B set theory into a theory modulo and provide it to ZenonModulo in order to verify the proof obligations of the BWare project.

4.3. Termination certificate verification

Termination is an important property to verify, especially in critical applications. Automated termination provers use more and more complex theoretical results and external tools (e.g. sophisticated SAT solvers) that make their results not fully trustable and very difficult to check. To overcome this problem, a language for termination certificates, called CPF, has been developed since several years now. Deducteam develops a formally certified tool, Rainbow, based on the Coq library CoLoR, that is able to automatically verify the correctness of such termination certificates.

5. New Software and Platforms

5.1. Software of the team

Deducteam develops several kinds of tools or libraries:

- Proof checkers:
  - Dedukti: proof checker for the $\lambda\Pi$-calculus modulo rewriting
  - Sukerujo: extension of Dedukti with syntactic constructions for records, strings, lists, etc.
  - Rainbow: CPF termination certificate verifier
- Tools for translating into Dedukti's proof format proofs coming from various other provers:
  - Coqine translates Coq proofs
  - Focalide translates Focalize proofs
  - Holide translates OpenTheory proofs (HOL-Light, HOL4, ProofPower)
  - Krajono translates Matita proofs
  - Sigmaid translates ς-calculus
- Automated theorem provers:
  - iProverModulo: theorem prover based on polarized resolution modulo
  - SuperZenon: extension of Zenon using superdeduction
  - ZenonArith: extension of Zenon using the simplex algorithm for arithmetic
  - ZenonModulo: extension of Zenon using deduction modulo and producing Dedukti proofs
  - Zipperposition: superposition prover featuring arithmetic and induction
  - HOT: automated termination prover for higher-order rewrite systems
  - Archsat: theorem prover using tableaux-like rules with a SAT core
- Libraries or generation tools:
  - CoLoR: Coq library on rewriting theory and termination
  - Logtk: library for first-order automated reasoning
  - mSat: modular SAT/SMT solver with proof output
  - Moca: generator of construction functions for types with relations on constructors

5.2. Novelties of the year

The main novelties this year are:

- CoLoR has been ported to Coq 8.5.
- F. Blanqui started to develop a prototype for developing Dedukti proofs interactively.

6. New Results

6.1. Dedukti

A. Assaf, G. Burel, R. Cauderlier, D. Delahaye, G. Dowek, C. Dubois, F. Gilbert, P. Halmagrand, O. Hermant, and R. Saillard, have finished writing a general presentation of the Dedukti system. This paper is submitted for publication.

Under the supervision of P. Halmagrand and G. Burel, D. Pham worked on the conversion of TSTP proof traces, as produced by automated theorem provers such as E, Zipperposition or Vampire, into Dedukti proofs. To that purpose, he modified Zenon modulo so that it reads TSTP files and tries to reprove the proof steps given by the trace.

R. Cauderlier defended his PhD thesis on the translation of programming languages to Dedukti and interoperability of proof systems [11]. He also presented his work on the use of Dedukti for rewriting-based proof transformation [15] and on the translation of FoCaLiZe in Dedukti [16]

6.2. Proof theory

G. Dowek and Y. Jiang have finished a paper on co-inductive and inductive complementation of inference systems. This paper is submitted for publication.

The paper of G. Dowek on the introduction of rules and derivations in a logic course has been published [24].

F. Gilbert has finished a paper on the automated constructivization of proofs, to appear in the proceedings of FOSSACS'17.

F. Thiré is working on the translation of the Fermat little theorem proof written in Matita to a proof written in HOL. A part of this work is developed in its internship report [25]. He is continuing this translation during his PhD thesis.

6.3. B Method

The B Method is a formal method mainly used in the railway industry to specify and develop safety-critical software. To guarantee the consistency of a B project, one decisive challenge is to show correct a large amount of proof obligations, which are mathematical formulas expressed in a classical set theory extended with a specific type system. To improve automated theorem proving in the B Method, Pierre Halmagrand proposes [17], [12] to use a first-order sequent calculus extended with a polymorphic type system, which is in particular the output proof-format of the tableau-based automated theorem prover Zenon. After stating some modifications of the B syntax and defining a sound elimination of comprehension sets, he proposes a translation of B formulas into a polymorphic first-order logic format. Then, he introduces the typed sequent calculus used by Zenon, and shows that Zenon proofs can be translated to proofs of the initial B formulas in the B proof system.

6.4. Termination

F. Blanqui revised his paper on “size-based termination of higher-order rewrite systems” submitted to the Journal of Functional Programming [23]. This paper is concerned with the termination, in Church’ simply-typed λ-calculus, of the combination of β-reduction and arbitrary user-defined rewrite rules fired using matching modulo α-congruence only. Several authors have devised termination criteria for fixpoint-based function definitions using deduction rules for bounding the size of terms inhabiting inductively defined types, where the size of a term is (roughly speaking) the set-theoretical height of the tree representation of its normal form. In the present paper, we extend this approach to rewriting-based function definitions and more general notions of size.

G. Dowek has finished writing a paper on the notion of model and its application to termination proofs for the $\lambda\Pi$-calculus modulo theory. This paper is submitted for publication.

6.5. Confluence

In $\lambda\Pi$modulo, congruences are expressed by rewrite rules that must enjoy precise properties, notably confluence, strong normalization, and type preservation. A difficulty is that these properties depend on each other in calculi of dependent types. To break the circularity, confluence is usually proved separately on untyped terms. A another difficulty then arises : computation do not terminate on untyped terms. A result of van Oostrom allows to show confluence of non-terminating left-linear higher-order rules, provided their critical pairs are development closed. This result was used for the encodings of HOL, Matita, and Coq up to version 8.4. Encoding the most recent version of Coq requires rules for universes that are confluent on open terms, while confluence on ground terms sufficed before. The encoding we recently developed for this new version of Coq has higher-order rules which are not left-linear, use pattern matching modulo associativity, commutativity and identity, and whose (joinable) critical pairs are not development closed. We have therefore developed a new powerful result for proving confluence of that sort of rules provided non-linear variables can only be instantiated by first-order expressions [18], [19].

6.6. Physics and computation

The paper of G. Dowek and P. Arrighi Free fall and cellular automata has been published [13]. As a sequel of this paper, G. Dowek and P. Arrighi have written a short note [22].

A. Díaz-Caro and G. Dowek have developed a new typing system for quantum λ-calculus allowing to distinguish between pure states and superpositions.

Under the supervision of S. Martiel and P. Arrighi, C. Chouteau worked on a particular notion of covariance in the model of causal graph dynamics. Causal graph dynamics are graph transformations constrained by Physics-inspired symmetries. The particular object of study of this internship was a restriction of this model to physical transformations of discrete geometrical spaces.

7. Partnerships and Cooperations

7.1. National Initiatives

7.1.1. ANR Locali

We are coordinators of the ANR-NFSC contract Locali with the Chinese Academy of Sciences.

7.1.2. ANR BWare

We are members of the ANR BWare, which started on September 2012 (David Delahaye is the national leader of this project). The aim of this project is to provide a mechanized framework to support the automated verification of proof obligations coming from the development of industrial applications using the B method. The methodology used in this project consists in building a generic platform of verification relying on different theorem provers, such as first-order provers and SMT solvers. We are in particular involved in the introduction of Deduction modulo in the first-order theorem provers of the project, i.e. Zenon and iProver, as well as in the backend for these provers with the use of Dedukti.

7.1.3. ANR Tarmac

We are members of the ANR Tarmac on models of computation, coordinated by Pierre Valarcher.

7.2. European Initiatives

7.2.1. Collaborations in European Programs, Except FP7 & H2020

Program: CA COST Action CA15123

Project acronym: EUTYPES

Project title: European research network on types for programming and verification

Duration: 21/03/16 - 20/03/20

Coordinator: Herman Geuvers

7.3. International Initiatives

7.3.1. Participation in Other International Programs

Login

Title: Logic and Information

International Partner (Institution - Laboratory - Researcher):

Universidad de Buenos Aires (Argentina) - Ricardo Oscar Rodrigues

Duration: 2015 - 2016

This project aims to propose an improvement on a long-term already existing collaboration between Inria, the brazilians and the argentin named team. We already have a CAPES-COFECUB cooperation (n. 690/10, namely “Teorias lógicas contemporâneas e a filosofia da linguagem: questões epistemológicas e semânticas”) that leaded to many students interchange and technical visits of Professors, including the organisation of some workshops (the last one was the 2nd Workshop on Logic and Semantics, at UERJ, Ilha Grande-RJ, Brazil. Prof. Gilles Dowek is also a Co-Advisor with Prof. Edward Hermann Haeusler of a brazilian Ph.D. Candidate in this project (and a former one also in this project, these two candidates finalised recently a sandwich doctorate - similar to stage doctorale - at Inria). Prof. Gilles Dowek also collaborates with other members of this team and is supervising a post-doc project of another member. Since 2011 members of the team presents.

FoQCoSS

Title: Foundations of Quantum Computation: Syntax and Semantics

International Partners (Institution - Laboratory - Researcher):

Universidad Nacional de Quilmes (Argentina) - Alejandro Diaz-Caro

CNRS (France) - Simon Perdrix

Duration: 2016 - 2017

The design of quantum programming languages involves the study of many characteristics of languages which can be seen as special cases of classical systems: parallelism, probabilistic systems, non-deterministic systems, type isomorphisms, etc. This project proposes to study some of these characteristics, which are involved in quantum programming languages, but also have a more immediate utility in the study of nowadays systems. In addition, from a more foundational point of view, we are interested in the implications of computer science principles for quantum physics. For example, the consequences of the Church-Turing thesis for Bell-like experiments: if some of the parties in a Bell-like experiment use a computer to decide which measurements to make, then the computational resources of an eavesdropper have to be limited in order to have a proper observation of non-locality. The final aim is to open a new direction in the search for a framework unifying computer science and quantum physics.

7.4. International Research Visitors

7.4.1. Internships

- Clément Chouteau, from May 2016 to July 2016
- David Pham (Univ. Évry) from June 2016 to July 2016

7.4.2. Visits to International Teams

7.4.2.1. Research Stays Abroad

F. Gilbert spent one month in the formal methods team at NASA Langley Research Center, to work with Cesar Munoz on the use of automated theorem provers to verify PVS proofs.

8. Dissemination

8.1. Promoting Scientific Activities

8.1.1. Scientific Events Organisation

8.1.1.1. General Chair, Scientific Chair

G. Dowek has co-organized the meeting Universality of Proofs in Dagstuhl.

8.1.1.2. Member of the Organizing Committees

G. Dowek is a member of the steering committee of FSCD.

8.1.2. Scientific Events Selection

8.1.2.1. Member of the Conference Program Committees

F. Blanqui was member of the program committee of the 2016 Coq Workshop.

8.1.2.2. Reviewer

F. Blanqui reviewed papers for IJCAR 2016 and CSL 2016.

8.1.3. Journal

8.1.3.1. Member of the Editorial Boards

G. Dowek is an editor of TCS-C.

8.1.4. Invited Talks

G. Dowek has been an invited speaker at ISEEP 2016.

G. Dowek has been an invited speaker at Physics and Computation 2016.

8.1.5. Scientific Expertise

G. Dowek has been a member of a commitee dedicated to an update of the high school informatics curriculum.

8.1.6. Research Administration

G. Dowek is the President of the Scientific Board of the Société informatique de France.

G. Dowek is a member of the Scientific Board of la Main à la Pâte.

G. Dowek is a member of the commission de réflexion sur l'éthique de la recherche en sciences et technologies du numérique d'Allistene.

G. Dowek is a member of the comité national français d'histoire et de philosophie des sciences et des techniques.

F. Blanqui is co-director of the pole 4 (programming: models, algorithms, languages and architectures) of Paris-Saclay University's doctoral school on computer science.

F. Blanqui is referent of LSV PhD students.

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

G. Dowek is attached professor at the École normale supérieure de Paris-Saclay. He has given a course at MPRI. He has given a course to the student preparing the teacher's recruiting exam Agrégation. He is responsible for the second year of master.

F. Blanqui gave a course (15h) on rewriting theory at the MPRI.

8.2.2. Supervision

PhD : Raphaël Cauderlier, Object-Oriented Mechanisms for Interoperability between Proof Systems, CNAM, 10/10/2016, Catherine Dubois

8.2.3. Juries

F. Blanqui was member of the 2016 Inria recruitment committee for young graduate scientists.

F. Blanqui was member of the jury for the best scientific production of the year within Paris-Saclay University's doctoral school on computer science.

9. Bibliography

Major publications by the team in recent years

[1] F. Blanqui. *Definitions by rewriting in the Calculus of Constructions*, in "Mathematical Structures in Computer Science", 2005, vol. 15, n^o 1, pp. 37-92 [*DOI :* 10.1017/S0960129504004426], http://hal.inria.fr/inria-00105648/en/

[2] F. Blanqui, A. Koprowski. *CoLoR: a Coq library on well-founded rewrite relations and its application to the automated verification of termination certificates*, in "Mathematical Structures in Computer Science", 2011, vol. 21, n^{o} 4, pp. 827-859, http://hal.inria.fr/inria-00543157/en/

[3] M. Boespflug. *Conception d'un noyau de vérification de preuves pour le lambda-Pi-calcul modulo*, École Polytechnique, 2011

[4] G. BUREL. *Experimenting with Deduction Modulo*, in "CADE 2011", V. SOFRONIE-STOKKERMANS, N. BJØRNER (editors), Lecture Notes in Artificial Intelligence, Springer, 2011, vol. 6803, pp. 162–176

[5] G. DOWEK. *Polarized Resolution Modulo*, in "IFIP Theoretical Computer Science", 2010

[6] G. DOWEK, T. HARDIN, C. KIRCHNER. *Theorem proving modulo*, in "Journal of Automated Reasoning", 2003, vol. 31, pp. 33-73

[7] C. DUBOIS, T. HARDIN, V. DONZEAU-GOUGE. *Building certified components within FOCAL*, in "Revised Selected Papers from the Fifth Symposium on Trends in Functional Programming, TFP 2004, München, Germany, 25-26 November 2004", H.-W. LOIDL (editor), Trends in Functional Programming, Intellect, 2006, vol. 5, pp. 33-48

[8] O. HERMANT. *Resolution is Cut-Free*, in "Journal of Automated Reasoning", March 2010, vol. 44, n^{o} 3, pp. 245-276

[9] M. JACQUEL, K. BERKANI, D. DELAHAYE, C. DUBOIS. *Verifying B Proof Rules using Deep Embedding and Automated Theorem Proving*, in "Software and Systems Modeling (SoSyM)", June 2013

[10] M. JACQUEL, K. BERKANI, D. DELAHAYE, C. DUBOIS. *Tableaux Modulo Theories Using Superdeduction*, in "Global Journal of Advanced Software Engineering (GJASE)", December 2014, vol. 1, pp. 1 - 13 [*DOI :* 10.1007/978-3-642-31365-3_26], https://hal.archives-ouvertes.fr/hal-01099338

Publications of the year

Doctoral Dissertations and Habilitation Theses

[11] R. CAUDERLIER. *Object-Oriented Mechanisms for Interoperability between Proof Systems*, Conservatoire National Des Arts et Métiers, Paris, October 2016, https://hal.inria.fr/tel-01415945

[12] P. HALMAGRAND. *Automated Deduction and Proof Certification for the B Method*, Conservatoire National Des Arts et Métiers, Paris, December 2016, https://hal.inria.fr/tel-01420460

Articles in International Peer-Reviewed Journals

[13] P. ARRIGHI, G. DOWEK. *Free fall and cellular automata*, in "Electronic Proceedings in Theoretical Computer Science", 2016, vol. 204, pp. 1 - 10 [*DOI :* 10.4204/EPTCS.204.1], https://hal.inria.fr/hal-01421712

International Conferences with Proceedings

[14] P. ARRIGHI, S. MARTIEL, S. PERDRIX. *Reversible Causal Graph Dynamics*, in "Reversible Computation", Bologna, Italy, Lecture Notes in Computer Science, July 2016, vol. 9720, pp. 73-88 [*DOI :* 10.1007/978-3-319-40578-0_5], https://hal.archives-ouvertes.fr/hal-01361427

[15] R. CAUDERLIER. *A Rewrite System for Proof Constructivization*, in "Workshop on Logical Frameworks and Meta-Languages: Theory and Practice 2016", Porto, Portugal, June 2016, pp. 1 - 7 [*DOI :* 10.1145/2966268.2966270], https://hal.inria.fr/hal-01420634

[16] R. Cauderlier, C. Dubois. *ML Pattern-Matching, Recursion, and Rewriting: From FoCaLiZe to Dedukti*, in "13th ICTAC International Colloquium on Theoretical Aspects of Computing", Taipei, Taiwan, October 2016, pp. 459 - 468 [*DOI :* 10.1007/978-3-319-46750-4_26], https://hal.inria.fr/hal-01420638

[17] P. Halmagrand. *Soundly Proving B Method Formulae Using Typed Sequent Calculus*, in "13th International Colloquium on Theoretical Aspects of Computing (ICTAC)", Taipei, Taiwan, A. Sampaio, F. Wang (editors), Lecture Notes in Computer Science, Springer International Publishing, October 2016, vol. 9965, pp. 196-213 [*DOI :* 10.1007/978-3-319-46750-4_12], https://hal.archives-ouvertes.fr/hal-01342849

Conferences without Proceedings

[18] A. Assaf, G. Dowek, J.-P. Jouannaud, J. Liu. *Encoding Proofs in Dedukti: the case of Coq proofs*, in "Proceedings Hammers for Type Theories", Coimbra, Portugal, Proc. Higher-Order rewriting Workshop, Easy Chair, July 2016, https://hal.inria.fr/hal-01330980

[19] A. Assaf, G. Dowek, J.-P. Jouannaud, J. Liu. *Untyped Confluence in Dependent Type Theories*, in "Proceedings Higher-Order Rewriting Workshop", Porto, Portugal, Proc. Higher-Order rewriting Workshop, Easy-Chair, June 2016, https://hal.inria.fr/hal-01330955

[20] K. Ji. *Resolution in Solving Graph Problems*, in "8th Working Conference on Verified Software: Theories, Tools, and Experiments (VSTTE 2016)", Toronto, Canada, July 2016, https://hal.archives-ouvertes.fr/hal-01245138

[21] S. Wang. *Higher Order Proof Engineering: Proof Collaboration, Transformation, Checking and Retrieval*, in "AITP 2016 - Conference on Artificial Intelligence and Theorem Proving", Obergurgl, Austria, April 2016, https://hal.inria.fr/hal-01250197

Other Publications

[22] P. Arrighi, G. Dowek. *What is the Planck constant the magnitude of?*, December 2016, working paper or preprint, https://hal.inria.fr/hal-01421711

[23] F. Blanqui. *Size-based termination of higher-order rewrite systems*, January 2017, working paper or preprint, https://hal.inria.fr/hal-01424921

[24] G. Dowek. *Rules and derivations in an elementary logic course*, January 2016, working paper or preprint, https://hal.inria.fr/hal-01252124

[25] F. Thiré. *Internship report MPRI 2 Reverse engineering on arithmetic proofs*, ENS Cachan ; Paris Diderot University, September 2016, 26 p. , https://hal.inria.fr/hal-01424816

References in notes

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

[27] D. Cousineau, G. Dowek. *Embedding Pure Type Systems in the lambda-Pi-calculus modulo*, in "Typed lambda calculi and applications", S. Ronchi della Rocca (editor), Lecture Notes in Computer Science, Springer-Verlag, 2007, vol. 4583, pp. 102-117

[28] D. DELAHAYE, D. DOLIGEZ, F. GILBERT, P. HALMAGRAND, O. HERMANT. *Proof Certification in Zenon Modulo: When Achilles Uses Deduction Modulo to Outrun the Tortoise with Shorter Steps*, in "IWIL - 10th International Workshop on the Implementation of Logics - 2013", Stellenbosch, South Africa, S. SCHULZ, G. SUTCLIFFE, B. KONEV (editors), EasyChair, December 2013, https://hal.inria.fr/hal-00909688

[29] D. DELAHAYE, D. DOLIGEZ, F. GILBERT, P. HALMAGRAND, O. HERMANT. *Zenon Modulo: When Achilles Outruns the Tortoise using Deduction Modulo*, in "LPAR - Logic for Programming Artificial Intelligence and Reasoning - 2013", Stellenbosch, South Africa, K. MCMILLAN, A. MIDDELDORP, A. VORONKOV (editors), Springer, December 2013, vol. 8312, pp. 274-290 [*DOI :* 10.1007/978-3-642-45221-5_20], https://hal.inria.fr/hal-00909784

[30] D. DELAHAYE, M. JACQUEL. *Recovering Intuition from Automated Formal Proofs using Tableaux with Superdeduction*, in "electronic Journal of Mathematics and Technology", February 2013, vol. 7, n^{o} 2, pp. 1 - 20, https://hal.archives-ouvertes.fr/hal-01099371

[31] R. GANDY. *Church's Thesis and Principles for Mechanisms*, in "The Kleene Symposium", North-Holland, 1980

[32] R. HARPER, F. HONSELL, G. PLOTKIN. *A Framework for Defining Logics*, in "Journal of the association for computing machinery", 1993, pp. 194–204

[33] J. HARRISON. *HOL Light: An Overview*, in "Theorem Proving in Higher Order Logics", S. BERGHOFER, T. NIPKOW, C. URBAN, M. WENZEL (editors), Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, vol. 5674, pp. 60-66, http://dx.doi.org/10.1007/978-3-642-03359-9_4

[34] M. JACQUEL, K. BERKANI, D. DELAHAYE, C. DUBOIS. *Verifying B Proof Rules using Deep Embedding and Automated Theorem Proving*, in "Software Engineering and Formal Methods", November 2011, vol. 7041, pp. 253-268 [*DOI :* 10.1007/978-3-642-24690-6_18], https://hal.archives-ouvertes.fr/hal-00722373

[35] M. JACQUEL, K. BERKANI, D. DELAHAYE, C. DUBOIS. *Tableaux Modulo Theories Using Superdeduction*, in "Global Journal of Advanced Software Engineering (GJASE)", December 2014, vol. 1, pp. 1 - 13 [*DOI :* 10.1007/978-3-642-31365-3_26], https://hal.archives-ouvertes.fr/hal-01099338

[36] K. KOROVIN. *iProver – An Instantiation-Based Theorem Prover for First-Order Logic (System Description)*, in "IJCAR", A. ARMANDO, P. BAUMGARTNER (editors), Lecture Notes in Artificial Intelligence, Springer, 2008, vol. 5195, pp. 292-298

[37] F. RABE, M. KOHLHASE. *A Scalable Module System*, in "Inf. Comput.", September 2013, vol. 230, pp. 1–54, http://dx.doi.org/10.1016/j.ic.2013.06.001