IN PARTNERSHIP WITH:
ENS Paris-Saclay

Activity Report 2019

Project-Team DEDUCTEAM

DEDUCTEAM

IN COLLABORATION WITH: Laboratoire specification et vérification (LSV)

RESEARCH CENTER
Saclay - Île-de-France

THEME
Proofs and Verification

Table of contents

Project-Team DEDUCTEAM

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

Keywords:

Computer Science and Digital Science:

A2.1.4. - Functional programming
A2.1.11. - Proof languages
A2.4.3. - Proofs
A3.1.1. - Modeling, representation
A7. - Theory of computation
A7.2. - Logic in Computer Science

Other Research Topics and Application Domains:

B7. - Transport and logistics

1. Team, Visitors, External Collaborators

Research Scientists

Gilles Dowek [Team leader, Inria, Senior Researcher, HDR]
Bruno Barras [Inria, Researcher]
Frédéric Blanqui [Inria, Researcher, HDR]
Valentin Blot [Inria, Researcher, from Mar 2019]
Jean-Pierre Jouannaud [University Paris-Saclay, Emeritus, HDR]

Faculty Members

Pablo Arrighi [Inria, Professor, from Sep 2019]
Guillaume Burel [ENSIIE, Associate Professor, until Aug 2019]

Post-Doctoral Fellows

Michael Farber [Inria, Post-Doctoral Fellow, from Sep 2019]
Rehan Malak [CNRS, Post-Doctoral Fellow, from Nov 2019]
Étienne Miquey [CNRS, Post-Doctoral Fellow, from Nov 2019]
Franck Slama [Inria, Post-Doctoral Fellow, until Aug 2019]

PhD Students

Mohamed Yacine El Haddad [CNRS, PhD Student]
Gaspard Ferey [École Nationale Supérieure des Mines de Paris , PhD Student]
Guillaume Genestier [École Normale Supérieure de Cachan, PhD Student]
Emilie Grienenberger [École Normale Supérieure de Cachan, PhD Student, from Oct 2019]
Gabriel Hondet [Inria, PhD Student, from Oct 2019]
François Thiré [École Normale Supérieure de Cachan, PhD Student, until Nov 2019]

Interns and Apprentices

Emilie Grienenberger [École Normale Supérieure de Cachan, from Mar 2019 until Aug 2019]
Gabriel Hondet [Inria, from Feb 2019 until Jul 2019]
Farzad Jafar Rahmani [Univ Paris-Saclay, from Mar 2019 until Aug 2019]
Houda Mouzoun [Mines ParisTech, from Jun 2019]
Jui Hsuan Wu [Inria, from Mar 2019 until Jul 2019]

Administrative Assistants

Adeline Lochet [Inria, Administrative Assistant, until May 2019]
Emmanuelle Perrot [Inria, Administrative Assistant]

External Collaborators

Guillaume Burel [ENSIIE, from Sep 2019]
Catherine Dubois [ENSIIE]
Olivier Hermant [École Nationale Supérieure des Mines de Paris]

2. Overall Objectives

2.1. Objectives

The project-team investigates the design of logical frameworks, in order to ensure interoperability between proof systems, and to the development of system-independent proof libraries. To achieve these goals, we develop

- a logical framework DEDUKTI, where several theories can be expressed,
- tools to import proofs developed in external proof systems to DEDUKTI theories,
- tools to translate proofs from one DEDUKTI theory to another,
- tools to export proofs expressed in DEDUKTI theories to an external proof system,
- tools to prove the confluence, the termination, and the consistency of theories expressed in DEDUKTI,
- tools to develop proofs directly in DEDUKTI,
- an encyclopedia LOGIPEDIA of proofs expressed in various DEDUKTI theories.

2.2. History

The idea that systems such as Euclidean geometry or set theory should be expressed, not as independent systems, but in a logical framework appeared with the design of the first logical framework: predicate logic, in 1928. Later, several more powerful logical frameworks have been designed: λ-prolog, Isabelle, the Edinburgh logical framework, Pure type systems, and Deduction modulo theory.

The logical framework that we use is a simple λ-calculus with dependent types and rewrite rules, called the $\lambda\Pi$-calculus modulo theory, and also the Martin-Löf logical framework, and it generalizes all the mentioned frameworks. It is implemented in the system DEDUKTI.

The first version of DEDUKTI was developed in 2011 by Mathieu Boespflug [24]. From 2012 to 2015, new versions of DEDUKTI were developed and several theories were expressed in DEDUKTI, allowing to import proofs developed in MATITA (with the tool KRAJONO), HOL LIGHT (with the tool HOLIDE), FOCALIZE (with the tool FOCALIDE), IPROVER, and ZENON, totalizing several hundred of megabytes of proofs.

From 2015 to 2018, we focused on the translation of proofs from one DEDUKTI theory to another and to the exporting of proofs to other proof systems. In particular the MATITA arithmetic library has been translated to a much weaker theory: constructive simple type theory, allowing to export it to COQ, LEAN, PVS, HOL LIGHT, and ISABELLE/HOL. This led us to develop, in 2018, an online proof encyclopedia LOGIPEDIA, allowing to share and browse this library. We also focused on the development of new theories in DEDUKTI, and on an interactive theorem prover on top of DEDUKTI.

3. Research Program

3.1. Logical Frameworks

A thesis, which is at the root of our research effort, is that logical systems should be expressed as theories in a logical framework. As a consequence, proof-checking systems should not be focused on one theory, such as Simple type theory, Martin-Löf's type theory, or the Calculus of constructions, but should be theory independent. On the more theoretical side, the proof search algorithms, or the algorithmic interpretation of proofs should not depend on the theory in which proofs are expressed, but this theory should just be a parameter. This is for instance expressed in the title of our invited talk at ICALP 2012: *A theory independent Curry-De Bruijn-Howard correspondence* [25].

Various limits of Predicate logic have led to the development of various families of logical frameworks: λ-prolog and Isabelle have allowed terms containing free variables, the Edinburgh logical framework has allowed proofs to be expressed as λ-terms, Pure type systems have allowed propositions to be considered as terms, and Deduction modulo theory has allowed theories to be defined not only with axioms, but also with computation rules.

The $\lambda\Pi$-calculus modulo theory, that is implemented in the system DEDUKTI and that is a synthesis of the Edinburgh logical framework and of Deduction modulo theory, subsumes them all. Part of our research effort is focused on improving the $\lambda\Pi$-calculus modulo theory, for instance allowing to define congruences with associative and commutative rewriting. Another part of our research effort is focused on the automatic analysis of theories to prove their confluence, termination, and consistency either by pencil and paper proofs or automatically [4].

3.2. Interoperability and proof encyclopediae

Using a single prover to check proofs coming from different systems naturally leads to investigate how these proofs can be translated from one theory to another and used in a system different from the system in which they have been developed. 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.

For each proof, independently of the system in which it has been developed, we should be able to identify the systems in which it can be expressed. For instance, we have shown that many proofs developed in the MATITA prover did not use the full strength of the logic of MATITA and could be exported, for instance, to the systems of the HOL family, that are based on a weaker logic.

Rather than importing proofs from one system, transforming them, and exporting them to another system, we can use the same tools to develop system-independent proof encyclopedia called Logipedia. In such a library, each proof is labeled with the theories in which it can be expressed and so with the systems in which it can be used.

3.3. Interactive theorem proving

If our main goal with DEDUKTI is to import, transform, and export proofs developed in other systems, we also want to investigate how DEDUKTI can be used as the basis of an interactive theorem prover. This leads to two new scientific questions: first, how much can a tactic system be theory independent, and then how does rewriting extends the possibility to write tactics.

This has led to the development of a new version of DEDUKTI, which supports metavariables. Several tactics have been developed for this system, which are intended to help a human user to write proofs in our system instead of writing proof terms by hand. This work is a continuation of the previous work the team did on DEMON, which was an extension of DEDUKTI, whereas the support for interactive theorem proving is now native in DEDUKTI.

4. Application Domains

4.1. Interoperability

Our main impact applications, for instance to proofs of programs, or to air traffic control, are through our cooperation with other teams.

As a matter of fact, we view our work on interoperability and on the design of a formal proof encyclopedia as a service to the formal proof community.

5. New Software and Platforms

5.1. Autotheo

KEYWORD: Automated deduction

SCIENTIFIC DESCRIPTION: Transformation of axiomatic theories into rewriting systems that can be used by iProverModulo.

FUNCTIONAL DESCRIPTION: Autotheo is a tool that transforms axiomatic theories into polarized rewriting systems, thus making them usable in iProverModulo. It supports several strategies to orient the axioms, some of them being proved to be complete, in the sense that ordered polarized resolution modulo the resulting systems is refutationally complete, some others being merely heuristics. In practice, Autotheo takes a TPTP input file and produces an input file for iProverModulo.

NEWS OF THE YEAR: Maintenance.

- Participant: Guillaume Burel
- Partner: ENSIIE
- Contact: Guillaume Burel
- Publication: Consistency Implies Cut Admissibility
- URL: http://www.ensiie.fr/~guillaume.burel/blackandwhite_autotheo.html.en

5.2. CoLoR

Coq Library on Rewriting and termination

KEYWORDS: Coq - Formalisation

FUNCTIONAL DESCRIPTION: CoLoR is a Coq library on rewriting theory and termination. It provides many definitions and theorems on various mathematical structures (quasi-ordered sets, relations, ordered semi-rings, etc.), data structures (lists, vectors, matrices, polynomials, finite graphs), term structures (strings, first-order terms, lambda-terms, etc.), transformation techniques (dependency pairs, semantic labeling, etc.) and (non-)termination criteria (polynomial and matrix interpretations, recursive path ordering, computability closure, etc.).

- Authors: Frédéric Blanqui and Sébastien Hinderer
- Contact: Frédéric Blanqui
- Publications: CoLoR: a Coq library on well-founded rewrite relations and its application to the automated verification of termination certificates - Automated Verification of Termination Certificates - CoLoR: a Coq library on rewriting and termination
- URL: http://color.inria.fr/

5.3. Coqine

Coq In dEdukti

KEYWORDS: Higher-order logic - Formal methods - Proof

FUNCTIONAL DESCRIPTION: CoqInE is a plugin for the Coq software translating Coq proofs into Dedukti terms. It provides a Dedukti signature file faithfully encoding the underlying theory of Coq (or a sufficiently large subset of it). Current development is mostly focused on implementing support for Coq universe polymorphism. The generated ouput is meant to be type-checkable using the latest version of Dedukti.

- Contact: Guillaume Burel
- URL: http://www.ensiie.fr/~guillaume.burel/blackandwhite_coqInE.html.en

5.4. Dedukti

KEYWORD: Logical Framework

FUNCTIONAL DESCRIPTION: Dedukti is a proof-checker for the LambdaPi-calculus modulo. As it can be parametrized by an arbitrary set of rewrite rules, defining an equivalence relation, this calculus can express many different theories. Dedukti has been created for this purpose: to allow the interoperability of different theories.

Dedukti's core is based on the standard algorithm for type-checking semi-full pure type systems and implements a state-of-the-art reduction machine inspired from Matita's and modified to deal with rewrite rules.

Dedukti's input language features term declarations and definitions (opaque or not) and rewrite rule definitions. A basic module system allows the user to organize his project in different files and compile them separately.

Dedukti features matching modulo beta for a large class of patterns called Miller's patterns, allowing for more rewriting rules to be implemented in Dedukti.

NEWS OF THE YEAR: There has been a new release 2.6 in 2018. This release provides a better control on module loading, and a better log of rewrite steps.

- Participants: François Thiré, Gaspard Ferey, Guillaume Genestier and Rodolphe Lepigre
- Contact: François Thiré
- Publications: Dedukti:un vérificateur de preuves universel - Rewriting Modulo β in the $\lambda\,\Pi$-Calculus Modulo - Expressing theories in the $\lambda\Pi$-calculus modulo theory and in the Dedukti system
- URL: https://deducteam.github.io/

5.5. Holide

KEYWORD: Proof

FUNCTIONAL DESCRIPTION: Holide translates HOL proofs to Dedukti[OT] proofs, using the OpenTheory standard (common to HOL Light and HOL4). Dedukti[OT] being the encoding of OpenTheory in Dedukti.

- Contact: Guillaume Burel
- URL: http://deducteam.gforge.inria.fr/holide/

5.6. HOT

Higher-Order Termination

FUNCTIONAL DESCRIPTION: HOT is an automated termination prover for higher-order rewriting, based on the notion of computability closure.

- Contact: Frédéric Blanqui
- URL: http://rewriting.gforge.inria.fr/hot.html

5.7. iProver Modulo

KEYWORDS: Automated deduction - Automated theorem proving

SCIENTIFIC DESCRIPTION: Integration of ordered polarized resolution modulo theory into the prover iProver.

FUNCTIONAL DESCRIPTION: iProver Modulo is an extension of the automated theorem prover iProver originally developed by Konstantin Korovin at the University of Manchester. It implements ordered polarized resolution modulo theory, a refinement of the resolution method based on deduction modulo theory. It takes as input a proposition in predicate logic and a clausal rewriting system defining the theory in which the formula has to be proved. Normalization with respect to the term rewriting rules is performed very efficiently through translation into OCaml code, compilation and dynamic linking. Experiments have shown that ordered polarized resolution modulo dramatically improves proof search compared to using raw axioms.

NEWS OF THE YEAR: Maintenance of Dedukti output

- Participant: Guillaume Burel
- Partner: ENSIIE
- Contact: Guillaume Burel
- Publications: A Shallow Embedding of Resolution and Superposition Proofs into the ??-Calculus Modulo - Experimenting with deduction modulo
- URL: https://github.com/gburel/iProverModulo

5.8. mSAT

KEYWORD: Propositional logic

FUNCTIONAL DESCRIPTION: mSAT is a modular, proof-producing, SAT and SMT core based on Alt-Ergo Zero, written in OCaml. The solver accepts user-defined terms, formulas and theory, making it a good tool for experimenting. This tool produces resolution proofs as trees in which the leaves are user-defined proof of lemmas.

- Contact: Guillaume Bury
- Publication: mSAT:An OCaml SAT Solver
- URL: https://github.com/Gbury/mSAT

5.9. Rainbow

Termination certificate verifier

KEYWORDS: Demonstration - Code generation - Verification

FUNCTIONAL DESCRIPTION: Rainbow is a set of tools for automatically verifying the correctness of termination certificates expressed in the CPF format used in the annual international competition of termination tools. It contains: a tool xsd2coq for generating Coq data types for representing XML files valid with respect to some XML Schema, a tool xsd2ml for generating OCaml data types and functions for parsing XML files valid with respect to some XML Schema, a tool for translating a CPF file into a Coq script, and a standalone Coq certified tool for verifying the correctness of a CPF file.

- Author: Frédéric Blanqui
- Contact: Frédéric Blanqui
- Publications: Automated verification of termination certificates - Automated verification of termination certificates
- URL: http://color.inria.fr/rainbow.html

5.10. Krajono

KEYWORD: Proof

FUNCTIONAL DESCRIPTION: Krajono translates Matita proofs into Dedukti[CiC] (encoding of CiC in Dedukti) terms.

- Contact: François Thiré

5.11. archsat

KEYWORDS: Automated theorem proving - First-order logic - Propositional logic

FUNCTIONAL DESCRIPTION: Archsat is an automated theorem prover aimed at studying the integration of first-order theorem prover technologies, such as rewriting, into SMT solvers.

- Contact: Guillaume Bury
- URL: https://gforge.inria.fr/projects/archsat

5.12. lrat2dk

KEYWORDS: Automated theorem proving - Proof

FUNCTIONAL DESCRIPTION: Take as input a SAT proof trace in LRAT format, which can be obtained from the de facto standard format DRAT using drat-trim. Output a proof checkable by Dedukti, in a shallow encoding of propositional logic.

- Participant: Guillaume Burel
- Partner: ENSIIE
- Contact: Guillaume Burel
- URL: https://github.com/gburel/lrat2dk

5.13. ekstrakto

KEYWORDS: TPTP - TSTP - Proof assistant - Dedukti

FUNCTIONAL DESCRIPTION: Extracting TPTP problems from a TSTP trace. Proof reconstruction in Dedukti from TSTP trace.

- Contact: Mohamed Yacine El Haddad
- URL: https://github.com/elhaddadyacine/ekstrakto

5.14. SizeChangeTool

KEYWORDS: Rewriting systems - Proof assistant - Termination

FUNCTIONAL DESCRIPTION: A termination-checker for higher-order rewriting with dependent types. Took part in the Termination Competition 2018 (http://termination-portal.org/wiki/Termination_Competition_2018) in the "Higher-Order Rewriting (union Beta)" category.

- Partner: Mines ParisTech
- Contact: Guillaume Genestier
- URL: https://github.com/Deducteam/SizeChangeTool

5.15. Logipedia

KEYWORDS: Formal methods - Web Services - Logical Framework

FUNCTIONAL DESCRIPTION: Logipedia is composed of two distinct parts: 1) A back-end that translates proofs expressed in a theory encoded in Dedukti to other systems such as Coq, Lean or HOL 2) A front-end that prints these proofs in a "nice way" via a website. Using the website, the user can search for a definition or a theorem then, download the whole proof into the wanted system.

Currently, the available systems are: Coq, Matita, Lean, PVS and OpenTheory. The proofs comes from a logic called STTForall.

In the long run, more systems and more logic should be added.

RELEASE FUNCTIONAL DESCRIPTION: This is the beta version of Logipedia. It implements the functionalities mentioned above.

- Contact: François Thiré
- URL: http://www.logipedia.science

6. New Results

6.1. Implementation of Dedukti

During his master internship with Frédéric Blanqui and Bruno Barras, Gabriel Hondet developed a new rewrite engine for Dedukti [22]. The algorithm used in the new rewriting engine is formalised and a correctness proof is provided. This algorithm is based on the pattern matching algorithm by Maranget and used in OCaml. It is extended to rewriting rules, λ terms and non linear patterns. Some interesting implementation details are evinced and then we compare the efficiency of the new engine to a naive matching algorithm and to the rewriting engine of Dedukti. The results show that our implementation handles large rewrite systems better that the naive algorithm, and is always better than Dedukti's.

During her internship with Frédéric Blanqui and Emilio Gallego, Houda Mouzoun developed a Dedukti plugin for the VSCode editor.

During his internship with Frédéric Blanqui and Valentin Blot, Jui-Hsuan Wu implemented a prototype algorithm for deciding whether a function defined by rewriting rules is injective or not [23], and also a new algorithm proposed by Frédéric Blanqui for checking that user-defined rewrite rules preserve typing.

Bruno Barras has developed a reduction machine implementing a strong call-by-need strategy for β-reduction and pattern-matching. Higher-order pattern-matching is not yet fully implemented. Regarding efficiency, an exponential speed-up can be observed compared to the current call-by-name implementation on a large class of examples, but a constant slow-down shows on examples where call-by-name is the optimal strategy. With Beniamino Accattoli, he has started studying the correctness of this machine, without pattern-matching. They proved that the machine correctly implements β-reduction, but have no result yet regarding the strategy or the time complexity.

6.2. Theory of $\lambda\Pi$-calculus modulo rewriting

Dependency pairs are a key concept at the core of modern automated termination provers for first-order term rewriting systems. In [14][15], Frédéric Blanqui, Guillaume Genestier and Olivier Hermant introduced an extension of this technique for a large class of dependently-typed higher-order rewriting systems. This improves previous results by Wahlstedt on the one hand and the first author on the other hand to strong normalization and non-orthogonal rewriting systems. This new result has been implemented in the termination-checker SizeChangeTool [17], which participated in the Termination Competition and is used by Dedukti.

During his internship with Frédéric Blanqui and Valentin Blot, Jui-Hsuan Wu designed an algorithm for deciding whether a function defined by rewriting rules is injective or not [23]. This allows to improve the unification algorithm used in Dedukti for inferring types and missing arguments.

The expressiveness of dependent type theory can be extended by identifying types modulo some additional computation rules. But, for preserving the decidability of type-checking or the logical consistency of the system, one must make sure that those user-defined rewriting rules preserve typing. Frédéric Blanqui has developed a new method to check that property using Knuth-Bendix completion. A prototype implementation by Jui-Hsuan Wu is available in Dedukti.

Confluence is a crucial property of rewriting. Gaspard Férey and Jean-Pierre Jouannaud formalized the higher-order rewriting relation on untyped terms implemented in Dedukti and studied various criteria to obtain confluence of higher-order rewrite systems considered together with beta. In particular Von Oostrom's decreasing diagrams technique is applied to multi-steps extensions of simple term rewriting to achieve confluence criteria based on the decidable computation of (orthogonal) higher-order critical pairs. This work assumes left-linearity of rules for now but current work aims at extending these techniques to prove confluence of non-left-linear rule restricted to subsets of terms [20].

Fran cois Thiré has worked on a criterion that would help proving metatheoretical results on Cumulative Type Systems, such as expansion postponment and the equivalence between typed and untyped presentations of conversion. This has been published andpresented at LFMTP'19 [19]

Frédéric Gilbert has written a preprint about the definition of proof certificates for predicative subtyping [21].

6.3. Proof reconstruction

Proof assistants often call automated theorem provers to prove subgoals. However, each prover has its own proof calculus and the proof traces that it produces often lack many details to build a complete proof. Hence these traces are hard to check and reuse in proof assistants. Dedukti is a proof checker whose proofs can be translated to various proof assistants: Coq, HOL, Lean, Matita, PVS. Yacine El Haddad, Guillaume Burel and Frédéric Blanqui implemented a tool Ektraskto [16] that extracts TPTP subproblems from a TSTP file and reconstructs complete proofs in Dedukti using automated provers able to generate Dedukti proofs like ZenonModulo or ArchSAT. This tool is generic: it assumes nothing about the proof calculus of the prover producing the trace, and it can use different provers to produce the Dedukti proof. We applied our tool on traces produced by automated theorem provers on the CNF problems of the TPTP library and we were able to reconstruct a proof for a large proportion of them, significantly increasing the number of Dedukti proofs that could be obtained for those problems.

Zenon Modulo and iProverModulo, two automated theorem provers that can produce Dedukti proofs, have been presented in an article accepted in the Journal of Automated Reasoning [12].

6.4. Translating proofs to Dedukti

Agda is a dependently-typed programming language developed at Chalmers University, Gothenburg, Sweden, for 20 years. Thanks to the propositions-as-types correspondence of Curry-Howard, Agda is often used as a proof-assistant. Guillaume Genestier developed with Jesper Cockx a prototypical translator from Agda to Dedukti, which supports well some of the mainly used features of Agda and translates hundreds of definitions of the standard libraries. This implementation led to new encodings of theories in Dedukti, regarding: Universe Polymorphism, Inductive and Record Types, Dependent Pattern Matching, eta convertibility. The implementation of this translator permits to improve both Agda and Dedukti. Indeed, we discovered some bugged (almost not used) functions in Agda and had to extend some existing functions to our purpose. On the Dedukti side, this implementation was the first usage of the newly implemented feature of rewriting modulo associativity and commutativity, which required some minor improvements. Furthermore, our translation of eta-expansion using a defined function led to an improvement in the verification of type preservation of rewriting rules in Dedukti.

Isabelle is a logical framework developed at Technical University of Munich and Cambridge University since the 90s. It implements several logics such as HOL and ZF and is used as part of large verification projects such as seL4 and Flyspeck. Gabriel Hondet developed with Makarius Wenzel (from Augsburg) an export from Isabelle propositions to Dedukti, which was later extended by Michael Färber and Makarius Wenzel to export proofs. This required substantial work on the Isabelle kernel to extend the reconstruction of proof terms based on the work of Stefan Berghofer. Our newly developed proof export allows for an independent verification of a substantial portion of the Isabelle/HOL standard library as well as for the integration of results proved in Isabelle into Logipedia.

6.5. Models of cubical type theory

Bruno Barras and Rehan Malak have developed further their Dedukti library of presheaves. Using this library, they have built a semi-simplicial model of System F.

6.6. A proof system for PCTL and CTL*

Gilles Dowek, Ying Jiang, and Wu Peng, have proposed a proof system for the probabilistic modal logic PCTL. A paper is in preparation.

Gilles Dowek, Ying Jiang, Wu Peng, and Wenhui Zhang have started to study a proof system for CTL*, that mixes constructive and classical aspects.

The article Towards Combining Model Checking and Proof Checking, of Ying Jiang, Jian Liu, Gilles Dowek, and Kailiang Ji, has been published in The Computer Journal [13].

6.7. System I

Gilles Dowek and Alejandro Díaz-Caro have defined a lambda-calculus, the system I, to represent the proofs of a variant minimal propositional logic where isomorphic propositions are identified. Their paper Proof Normalisation in a Logic Identifying Isomorphic Propositions, has been presented at the International Conference on Formal Structures for Computation and Deduction. A second paper The virtues of eta-expansion in System I, showing that the addition of eta-expansion to system I actually simplies the system has been submitted to publication.

6.8. Computing with global environments

The call-by-need evaluation strategy for the λ-calculus is an evaluation strategy that lazily evaluates arguments only if needed, and if so, shares computations across all places where it is needed. To implement this evaluation strategy, abstract machines require some form of global environment. While abstract machines usually lead to a better understanding of the flow of control during the execution, easing in particular the definition of continuation-passing style translations, the case of machines with global environments turns out to be much more subtle.

In collaboration with Hugo Herbelin, Étienne Miquey introduced F_Υ, a calculus featuring a data type for typed stores and a mechanism of explicit coercions witnessing store extensions. This calculus defines a generic target of typed continuationand-environment-passing style translations for several calculi with global environment: it is compatible with different evaluation strategy (call-by-need, call-by-name, call-by-value) and different type systems (simple types, system F). On the logical side, these translations broadly amounts to a Kripke forcing-like translation mixed with a negative translation (for the continuation-passing part).

6.9. Computional interpretation of the axiom scheme of comprehension

The axiom scheme of comprehension is the cornerstone of second-order arithmetic, a logical theory in which most of mathematics can be formalized. Historically, comprehension was obtained from the negative translation of the axiom of choice, this axiom being interpreted by bar recursion. This led to cluttered and inefficient interpretations of second-order arithmetic.

Valentin Blot simplified this interpretation by proving that the axiom scheme of comprehension has a direct computational interpretation through a variant of bar recursion called update recursion. This new interpretation leads to a more efficient computational interpretation of proofs in second-order arithmetic, and paves the way for a convergence of the two existing interpretations: bar recursion and System F.

6.10. Alignment of logical connectives

Émilie Grienenberger and Gilles Dowek have studied in practice the alignment of logical connectives between proofs systems, a first step towards concept alignement, by the export of the HOL Light standard library using axiomatized connectives to Dedukti. More theoretically, an ecumenical system–where classical and intuitionistic logics coexist–was introduced to act as an exchange platform between proof systems.

6.11. Quantum Computing

The article Two linearities for quantum computing in the lambda calculus, of Alejandro Díaz-Caro, Gilles Dowek, and Juan Pablo Rinaldi, first published in the proceedings of Theory and Practice of Natural Computing 2017, has been published in the journal Biosystems.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

Valentin Blot obtained with Chantal Keller funding for a 4-year project involving a PhD student, a research engineer (2 years) and a post-doctoral researcher (2 years). This funding is part of the Inria - Nomadic labs partnership for Tezos blockchain.

8. Partnerships and Cooperations

8.1. Regional Initiatives

Valentin Blot obtained funding for hiring Étienne Miquey as a post-doctoral researcher from Île-de-France region's DIM-RFSI (Domaine d'Intérêt Majeur - Réseau Francilien en Sciences Informatiques).

8.2. National Initiatives

The ANR PROGRAMme is an ANR for junior researcher Liesbeth Demol (CNRS, UMR 8163 STL, University Lille 3) to which G. Dowek participates. The subject is: "What is a program? Historical and Philosophical perspectives". This project aims at developing the first coherent analysis and pluralistic understanding of "program" and its implications to theory and practice.

8.3. International Initiatives

8.3.1. Inria International Partners

8.3.1.1. Informal International Partners

Frédéric Blanqui cooperates with various researchers in Japan: Makato Hamana (Gunma University), Yoji Akama (Tohoku University) and Kentaro Kikuchi (Tohoku University).

8.4. International Research Visitors

8.4.1. Visits to International Teams

8.4.1.1. Research Stays Abroad

Gilles Dowek has spent two weeks at the Institute of Software in Beijing where he has worked with Ying Jiang, Wu Peng, and Wenhui Zhang.

Gilles Dowek has spent two weeks at the University of Buenos Aires where he has worked with Alejandro Díaz-Caro.

Frédéric Blanqui has been invited for two weeks in Japan by Yoji Akama (Tohoku University) and Makato Hamana (Gunma University).

As a "Short Term Scientific Mission" financed by COST Action EUTypes, Guillaume Genestier spent five weeks in Chalmers University, Gothenburg, Sweden, to cooperate with Jesper Cockx and Andreas Abel on the translation between the proof assistant Agda and Dedukti.

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific Events: Organisation

9.1.1.1. Member of the Organizing Committees

Deducteam has organized the kick off meeting of Logipedia on 21-23 January 2019 in Cachan.

Frédéric Blanqui and Olivier Hermant organized the 11th International School on Rewriting (ISR) in Paris on 1-6 July 2019. The school offered to 37 participants from 8 countries, 69 hours of lectures in two parallel sessions, given by 15 lecturers from 6 countries.

Valentin Blot co-organized the Facets of realizability workshop on 1-3 July 2019, bringing together 22 researchers from 6 countries.

Frédéric Blanqui is Workshop Chair of the ACM/IEEE Symposium on Logic in Computer Science (LICS).

9.1.2. Scientific Events: Selection

9.1.2.1. Member of the Conference Program Committees

Gilles Dowek has been a member of the program committee of the Nasa Formal Methods Symposium (NFM'19) and Interactive Theorem Proving (ITP'19).

Frédéric Blanqui has been member of the program committee of the 16th International Colloquium on Theoretical Aspects of Computing (ICTAC'19), the 6th Workshop on Proof eXchange for Theorem Proving (PxTP'19), and of Large Mathematical Libraries (LML'19).

Valentin Blot was a member of the program committee for the french conference "Journées Francophones des Langages Applicatifs" 2020.

9.1.2.2. Reviewer

Guillaume Burel reviewed articles submitted to PxTP'19 and LOPSTR'19.

Valentin Blot reviewed articles submitted to LICS'19.

Bruno Barras reviewed an article submitted to CSL'20.

9.1.3. Journal

9.1.3.1. Reviewer - Reviewing Activities

Guillaume Burel reviewed an article for Logical Methods in Computer Science.

Bruno Barras reviewed articles submitted to the Annals of Mathematics and Artificial Intelligence and Mathematical Structures in Computer Science

9.1.4. Invited Talks

Gilles Dowek has been invited to the meeting Big proofs in Edinburgh, where he has given a talk on Logipedia.

Gilles Dowek has participated to the meeting of the Proof society in Swansea where he has given a talk on Logipedia.

Gilles Dowek has given talk on Logipedia in Lille, Strasbourg, Beijing, Rio de Janeiro, and in the virtual "Laboratoire International de Recherche en Informatique et Mathématiques Appliquées".

Frédéric Blanqui gave talks at Gunma University (Japan) and Tohoku University (Japan).

Guillaume Genestier gave a seminar at the logic group of Chalmers University (Sweden).

Valentin Blot gave an invited talk at a realizability workshop on 18-20 September 2019 in Marseille, and a talk at the IRIF semantics working group.

9.1.5. Leadership within the Scientific Community

Frédéric Blanqui is member of the Steering Committees of LICS, TYPES and ISR.

9.1.6. Scientific Expertise

Bruno Barras reviewed a research proposal submitted to the Dutch National Research Organization (NWO).

9.1.7. Research Administration

Frédéric Blanqui is co-director of the pole 4 of the doctoral school STIC of the University Paris-Saclay.

Frédéric Blanqui is member of the committee of the doctoral school of the ENS Paris-Saclay.

Frédéric Blanqui is in charge of following PhD students at LSV.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

- Master: Frédéric Blanqui, formal languages, 21h, M1, ENSIIE, France
- Master: Frédéric Blanqui, rewriting theory, 14h, M1, ENS Paris-Saclay, France
- Master: Frédéric Blanqui, λ-calculus and theories in first-order logic, 18h, M1/M2, ENS Paris-Saclay, France
- Master: Bruno Barras, Proof Assistants, 12h, M2, MPRI, France
- Master: Gabriel Hondet, rewriting theory TD, 14h, M!, ENS Paris-Saclay, France
- Licence: Gabriel Hondet, operating systems and architecture TD, 22.5h, L3, ENS Paris-Saclay, France
- Licence: Gabriel Hondet, programming TD, 12h, L3, ENS Paris-Saclay, France
- License: Gaspard Férey, Théorie des Langages, 44h, L3, EISTI
- IUT: Yacine El Haddad, Programmation Web Côté Serveur, 62h, IUT of Orsay

9.2.2. Supervision

- PhD in progress: Guillaume Genestier, termination in $\lambda\Pi$-calculus modulo theory, 01/10/17, Frédéric Blanqui and Olivier Hermant.
- PhD in progress: Mohamed Yacine El Haddad, using automated provers in proof assistants, 05/01/18, Frédéric Blanqui and Guillaume Burel.
- PhD in progress: Gabriel Hondet, translating PVS proofs to Dedukti, 01/10/19, Frédéric Blanqui and Gilles Dowek.

9.2.3. Juries

Bruno Barras and Valentin Blot have participated to the PhD jury of Youssef El Bakouny from Université St-Joseph, Beirut (Lebanon).

9.3. Popularization

9.3.1. Articles and contents

Articles

- The synthesis article Algorithmes et modèles : l'histoire d'une convergence, written a decade ago has been published in Pierre Mounoud, Leçons de mathématiques d'aujourd'hui : Volume 5 (Cassini,2019).

Books

- Gilles Dowek, Ce dont on ne peut parler, il faut l'écrire : Langues et langages (Le Pommier, 2019)

Manuscripts

- Gilles Dowek, How the physical Church-Turing thesis changed the concept of machine, manuscript, 2019.
- Gilles Dowek, Two consequences of the hypothesis that we are within the world, manuscript, 2019.
- Gilles Dowek, Instinct, language, and artificial intelligence, manuscript, 2019.

9.3.2. Interventions

Gilles Dowek has been invited to the PROGRAMme workshop "Machines" where he has presented a paper How the physical Church-Turing thesis changed the concept of machine.

Gilles Dowek has been invited to the Conference on Robotics, AI, and Humanity, Science, Ethics, and Policy organized by the Pontifical Academy of Sciences and the Pontifical Academy of Social Sciences where he has presented a paper Instinct, language, and artificial intelligence.

Gilles Dowek has been invited at the meeting Le hasard, le calcul et la vie, in Cerisy, where he has given a talk Un Chaos discret.

Gilles Dowek has been invited to the meeting Experiencing reality directly in Jerusalem where he has presented a paper Two consequences of the hypothesis that we are within the world.

10. Bibliography

Major publications by the team in recent years

[1] A. ASSAF, G. BUREL, R. CAUDERLIER, D. DELAHAYE, G. DOWEK, C. DUBOIS, F. GILBERT, P. HALMAGRAND, O. HERMANT, R. SAILLARD. *Expressing theories in the λΠ-calculus modulo theory and in the Dedukti system*, in "22nd International Conference on Types for Proofs and Programs, TYPES 2016", Novi SAd, Serbia, May 2016, https://hal-mines-paristech.archives-ouvertes.fr/hal-01441751

[2] B. BARRAS, T. COQUAND, S. HUBER. *A generalization of the Takeuti-Gandy interpretation*, in "Mathematical Structures in Computer Science", 2015, vol. 25, n° 5, pp. 1071–1099, https://doi.org/10.1017/S0960129514000504

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

[4] F. BLANQUI, J.-P. JOUANNAUD, A. RUBIO. *The Computability Path Ordering*, in "Logical Methods in Computer Science", October 2015 [*DOI :* 10.2168/LMCS-11(4:3)2015], https://hal.inria.fr/hal-01163091

[5] V. BLOT. *An interpretation of system F through bar recursion*, in "32nd ACM/IEEE Symposium on Logic in Computer Science", IEEE, 2017

[6] G. BUREL, G. BURY, R. CAUDERLIER, D. DELAHAYE, P. HALMAGRAND, O. HERMANT. *First-Order Automated Reasoning with Theories: When Deduction Modulo Theory Meets Practice*, in "Journal of Automated Reasoning", 2019 [*DOI :* 10.1007/s10817-019-09533-z], https://hal.archives-ouvertes.fr/hal-02305831

[7] D. COUSINEAU, G. DOWEK. *Embedding Pure Type Systems in the λΠ-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

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

[9] O. HERMANT. *Resolution is Cut-Free*, in "Journal of Automated Reasoning", March 2010, vol. 44, n° 3, pp. 245-276

[10] 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

[11] 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

Articles in International Peer-Reviewed Journals

[12] G. BUREL, G. BURY, R. CAUDERLIER, D. DELAHAYE, P. HALMAGRAND, O. HERMANT. *First-Order Automated Reasoning with Theories: When Deduction Modulo Theory Meets Practice*, in "Journal of Automated Reasoning", 2019, forthcoming [*DOI :* 10.1007/s10817-019-09533-z], https://hal.archives-ouvertes.fr/hal-02305831

[13] Y. JIANG, J. LIU, G. DOWEK, K. JI. *Towards Combining Model Checking and Proof Checking*, in "The Computer Journal", 2019, forthcoming [*DOI :* 10.1093/COMJNL/BXY112], https://hal.inria.fr/hal-01970274

International Conferences with Proceedings

[14] F. BLANQUI, G. GENESTIER, O. HERMANT. *Dependency Pairs Termination in Dependent Type Theory Modulo Rewriting*, in "FSCD 2019 - 4th International Conference on Formal Structures for Computation and Deduction", Dortmund, Germany, June 2019, https://arxiv.org/abs/1906.11649 [*DOI :* 10.4230/LIPIcs.FSCD.2019.9], https://hal.inria.fr/hal-01943941

[15] F. BLANQUI, G. GENESTIER, O. HERMANT. *Dependency Pairs Termination in Dependent Type Theory Modulo Rewriting*, in "TYPES 2019 - 25th International Conference on Types for Proofs and Programs", Oslo, Norway, M. BEZEM, N. VAN DER WEIDE (editors), June 2019, pp. 30-31, https://hal.archives-ouvertes.fr/hal-02442484

[16] M. Y. EL HADDAD, G. BUREL, F. BLANQUI. *Ekstrakto A tool to reconstruct Dedukti proofs from TSTP files (extended abstract)*, in "PxTP 2019 - Sixth Workshop on Proof eXchange for Theorem Proving", Natal, Brazil, Electronic Proceedings in Theoretical Computer Science, August 2019, vol. 301, pp. 27-35 [*DOI :* 10.4204/EPTCS.301.5], https://hal.inria.fr/hal-02200548

[17] G. GENESTIER. *SizeChangeTool: A Termination Checker for Rewriting Dependent Types*, in "HOR 2019 - 10th International Workshop on Higher-Order Rewriting", Dortmund, Germany, M. AYALA-RINCÓN, S. GHILEZAN, J. G. SIMONSEN (editors), Joint Proceedings of HOR 2019 and IWC 2019, June 2019, pp. 14-19, https://hal.archives-ouvertes.fr/hal-02442465

National Conferences with Proceedings

[18] S. COLIN, R. LEPIGRE, G. SCHERER. *Unboxing Mutually Recursive Type Definitions in OCaml*, in "JFLA 2019 - 30 èmes journées francophones des langages applicatifs", Les Rousses, France, January 2019, https://arxiv.org/abs/1811.02300 , https://hal.inria.fr/hal-01929508

Conferences without Proceedings

[19] F. THIRÉ. *Cumulative Types Systems and Levels*, in "LFMTP 2019 - Logical Frameworks and Meta-Languages: Theory and Practice", Vancouver, Canada, June 2019, https://hal.archives-ouvertes.fr/hal-02150179

Other Publications

[20] G. FEREY, J.-P. JOUANNAUD. *Confluence in (Un)Typed Higher-Order Theories by means of Critical Pairs*, December 2019, working paper or preprint, https://hal.archives-ouvertes.fr/hal-02096540

[21] F. GILBERT. *Verifiable certificates for predicate subtyping*, January 2019, working paper or preprint, https://hal.inria.fr/hal-01977585

[22] G. HONDET. *Efficient rewriting using decision trees*, ENAC ; Univesité Toulouse III- Paul Sabatier, August 2019, https://hal.inria.fr/hal-02317471

[23] J.-H. WU. *Checking the type safety of rewrite rules in the λΠ-calculus modulo rewriting*, Ecole Normale Supérieure, September 2019, https://hal.inria.fr/hal-02288720

References in notes

[24] M. BOESPFLUG. *Conception d'un noyau de vérification de preuves pour le λΠ-calcul modulo*, École Polytechnique, 2011

[25] G. DOWEK. *A Theory Independent Curry-de Bruijn-howard Correspondence*, in "Proceedings of the 39th International Colloquium Conference on Automata, Languages, and Programming - Volume Part II", Berlin, Heidelberg, ICALP'12, Springer-Verlag, 2012, pp. 13–15, http://dx.doi.org/10.1007/978-3-642-31585-5_2