Activity Report 2013

Exploratory Action DEDUCTEAM

Deduction modulo, interopérabilité et démonstration automatique

RESEARCH CENTER
Paris - Rocquencourt

Table of contents

Exploratory Action DEDUCTEAM

Keywords: Type Systems, Proof Theory, Automated Theorem Proving, Model Of Computation, Safety

Creation of the Team: 2011 December 01*, updated into exploratory Action:* 2013 January 01.

1. Members

Research Scientists

Gilles Dowek [Team leader, Inria, Senior Researcher, HdR]
Frédéric Blanqui [Inria, Researcher, from Sep 2013, HdR]
Catherine Dubois [Inria until Aug 2013, Ensiie from Sep 2013, delegation Inria, HdR]

External Collaborators

Guillaume Burel [Ensiie]
David Delahaye [Cnam, HdR]
Alejandro Díaz-Caro [ATER Univ. Paris X]
Olivier Hermant [Institut Mines Télécom]

PhD Students

Ali Assaf [École Polytechnique]
Raphaël Cauderlier [Ens Cachan]
Simon Cruanes [École Polytechnique]
Pierre Halmagrand [CNAM, from Apr 2013]
Kailiang Ji [Inria, granted by ANR LOCALI project]
Kim-Quyen Ly [UJF, granted by LIAMA, from Sep 2013]
Pierre Néron [École Polytechnique, until Aug 2013]
Ronan Saillard [Institut Mines Télécom]

Post-Doctoral Fellows

Hugo Dos Santos Macedo [Inria, from Feb 2013]
Benoît Valiron [from Mar 2013 to Sep 2013]

Visiting Scientists

Edward Hermann Haeusler [PUC Rio, from Oct 2013]
Ying Jiang [Institute of software, Chinese Academy of Sciences]
Bruno Lopes Vieira [PUC Rio, from Mar 2013 to Jul 2013]
Cecilia Reis Englander Lustosa [PUC Rio, until Jan 2013]

Administrative Assistant

Hélène Milome [Inria]

Others

Frédéric Gilbert [Min. Écologie, Intern, from Mar 2013 to Aug 2013]
Frédéric Lang [ENSTA, Intern, from Apr 2013 to Jun 2013]

2. Overall Objectives

2.1. Objectives

The team investigates applications of recent results in proof theory to the design of proof checkers and automated theorem proving systems. It develops the Dedukti proof checker and the iProver modulo automated theorem proving system.

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.

2.3. Highlights of the Year

The Version 2 of Dedukti has been released. Gilles Dowek has been invited speaker to CSR, Hapoc, and to the Colloquium of the University Pierre et Marie Curie. David Delahaye has been an invited speaker of PSATTT.

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 [46]. It was shown in 2007, that extending the simplest λ-calculus with dependent types, the $\lambda\Pi$-calculus, with an equivalence relation, 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 [43], HOL [50], 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, 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 [53].

A thesis, which is at the root of our research effort, and which was already formulated by the team of the Logical Framework [49] 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*.

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 [6] and it could then be implemented. This implementation that builds on the iProver effort [52] 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 *Super Zenon* [5] and is an extension to superdeduction (which is a variant of Deduction modulo), and the second one is called *Zenon Modulo* [22], [23] 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 *Super Zenon* has been successfully applied to verify *B* proof rules of *Atelier B* (work in collaboration with *Siemens*). Similarly, we plan to apply *Zenon Modulo* 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 [48] 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. Tools for proofs in B

Set theory appears to be an appropriate theory for automated theorem provers based on Deduction modulo, in particular the several extensions of *Zenon* (*Super Zenon* and *Zenon Modulo*). 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* [10]. From this work presented in the Doctoral dissertation of Mélanie Jacquel, the *Super Zenon* tool [5] 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 *Zenon Modulo* [22], [23] (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 *Zenon Modulo* in order to verify the proof obligations of the *BWare* project.

5. Software and Platforms

5.1. Dedukti

Dedukti is a proof-checker for the $\lambda\Pi$-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 [42] for type-checking semi-full pure type systems and implements a state-of-the-art reduction machine inspired from Matita's [40] 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 its project in different files and compile them separately.

Dedukti has been developed by Mathieu Boespflug, Olivier Hermant, Quentin Carbonneaux, and Ronan Saillard. It is composed of about 1000 lines of OCaml.

5.2. Coqine, Holide and Focalide

Dedukti comes with three companion tools: Holide, an embedding of HOL proofs through the OpenTheory format [51], Coqine, an embedding of Coq proofs, and Focalide, an embedding of FoCaLiZe certified programs. All of the OpenTheory standard library and a part of Coq's and FoCaLiZe's libraries are checked by Dedukti.

A preliminary version of Coqine supports the following features of Coq: the raw Calculus of Constructions, inductive types, and fixpoint definitions. Coqine is currently being rewritten to support universes. Coqine has been developed by Mathieu Boespflug, Guillaume Burel, and Ali Assaf.

Holide supports all the features of HOL, including ML-polymorphism, constant definitions, and type definitions. It is able to translate all of the OpenTheory standard theory library. Holide has been developed by Ali Assaf.

Focalide supports the object-oriented features of FoCaLiZe, including inheritance, late-binding, redefinition and class parameters, and functional programming features of FoCaLiZe. It has been updated to work with the last version of FoCaLiZe. Focalide has been developed by Raphaël Cauderlier.

5.3. iProver Modulo

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, a refinement of the Resolution method based on Deduction modulo. 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. iProver modulo is also able to produce proofs that can be checked by Dedukti, therefore improving confidence. iProver modulo is written in OCaml, it consists of 1,200 lines of code added to the original iProver.

A tool that transforms axiomatic theories into polarized rewriting systems, thus making them usable in iProver Modulo, has also been developed. Autotheo 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 transforms the axioms into rewriting rules, and produces an input file for iProver Modulo.

iProver Modulo and autotheo have been developed by Guillaume Burel. iProver Modulo is released under a GPL license.

iProver Modulo entered CASC-24, the competition of Automated Theorem Provers held during the 24th CADE conference, in the first-order theorem division, using autotheo to orient the axioms of the problems.

5.4. Super Zenon and Zenon Modulo

Several extensions of the *Zenon* automated theorem prover (developed by Damien Doligez at *Inria* in the *Gallium* team) to Deduction modulo have been studied. These extensions intend to be applied in the context of the automatic verification of proof rules and obligations coming from industrial applications formalized using the *B* method.

The first extension, developed by Mélanie Jacquel and David Delahaye, is called *Super Zenon* and is an extension of *Zenon* to superdeduction, which can be seen as a variant of Deduction modulo. This extension is a generalization of previous experiments [10] together with Catherine Dubois and Karim Berkani (*Siemens*), where *Zenon* has been used and extended to superdeduction to deal with the *B* set theory and automatically prove proof rules of *Atelier B*. This generalization consists in allowing us to apply the extension of *Zenon* to superdeduction to any first order theory by means of a heuristic that automatically transforms axioms of the theory into rewrite rules. This work is described in [5], which also proposes a study of the possibility of recovering intuition from automated proofs using superdeduction.

The second extension, developed by Pierre Halmagrand, David Delahaye, Damien Doligez, and Olivier Hermant, is called *Zenon Modulo* and is an extension of *Zenon* to Deduction modulo. Compared to *Super Zenon*, this extension allows us to deal with rewrite rules both over propositions and terms. Like *Super Zenon*, *Zenon Modulo* is able to deal with any first order theory by means of a similar heuristic. To assess the approach of *Zenon Modulo*, we have applied this extension to the first order problems coming from the TPTP library. An increase of the number of proved problems has been observed, with in particular a significant increase in the category of set theory. This result in the set category allows us to be quite optimistic for the use of *Zenon Modulo* in the framework of the *BWare* project, since the *B* method is actually based on a set theory modeling technique. Over these problems of the TPTP library, we have also observed a significant proof size reduction, which confirms this aspect of Deduction modulo. These results are gathered into two publications [22], [23].

5.5. Zipperposition and Logtk

Zipperposition is an implementation of the superposition method. It experiments theory handling using its extensibility features. Current development includes splitting it into a generic library for representing logic data structures and algorithms, and a prover that uses this library. The library is called LOGTK("logic tool kit"). Zipperposition itself, in its development version, can deal with polymorphic logic, and integer and rational arithmetic. Theoretical work on an efficient inference system for arithmetic is ongoing. It entered CASC-24, the competition of Automated Theorem Provers held during the 24th CADE conference, in the first-order theorem division.

Zipperposition is developed by Simon Cruanes.

Logtk is in active development, in parallel with Zipperposition, and an unstable version is released here. The library, among other things, provides first-order terms, with polymorphic types and some type inference, first-order formulas, unification, term ordering, term rewriting, reduction of formulas to CNF, congruence closure, and an optional implementation of the meta-prover Simon Cruanes and Guillaume Burel have published about

[21]. Logtk focuses on efficiency and generality of its constructs. Several term indexing structures usable for rewriting, resolution, or subsumption checking expose a functorial interface that allows to associate any data with indexed terms.

Logtk also relies on some smaller OCaml developments by Simon Cruanes, especially a full-fledged implementation of Datalog and efficient iterators.

5.6. CoLoR

CoLoR is a Coq library on rewriting theory and termination of more than 83,000 lines of code [2]. It provides definitions and theorems for:

- Mathematical structures: relations, (ordered) semi-rings.
- Data structures: lists, vectors, polynomials with multiple variables, finite multisets, matrices, finite graphs.
- Term structures: strings, algebraic terms with symbols of fixed arity, algebraic terms with varyadic symbols, pure and simply typed λ-terms.
- Transformation techniques: conversion from strings to algebraic terms, conversion from algebraic to varyadic terms, arguments filtering, rule elimination, dependency pairs, dependency graph decomposition, semantic labelling.
- Termination criteria: polynomial interpretations, multiset ordering, lexicographic ordering, first and higher order recursive path ordering, matrix interpretations.

CoLoR is distributed under the CeCILL license. It is currently developed by Frédéric Blanqui and Kim-Quyen Ly, but various people participated to its development since 2006 (see the website for more information).

5.7. HOT

HOT is an automated termination prover for higher-order rewrite systems based on the notion of computability closure and size annotation [44]. It won the 2012 competition in the category "higher-order rewriting union beta". The sources are not public. It is developed by Frédéric Blanqui.

5.8. Moca

Moca is a construction functions generator for OCaml data types with invariants.

It allows the high-level definition and automatic management of complex invariants for data types. In addition, it provides the automatic generation of maximally shared values, independently or in conjunction with the declared invariants.

A relational data type is a concrete data type that declares invariants or relations that are verified by its constructors. For each relational data type definition, Moca compiles a set of construction functions that implements the declared relations.

Moca supports two kinds of relations:

- predefined algebraic relations (such as associativity or commutativity of a binary constructor),
- user-defined rewrite rules that map some pattern of constructors and variables to some arbitrary user's define expression.

The properties that user-defined rules should satisfy (completeness, termination, and confluence of the resulting term rewriting system) must be verified by a programmer's proof before compilation. For the predefined relations, Moca generates construction functions that allow each equivalence class to be uniquely represented by their canonical value.

Moca is distributed under QPL. It is developed by Frédéric Blanqui, Pierre Weis (EPI Pomdapi) and Richard Bonichon (CEA).

5.9. Rainbow

Rainbow is a tool for automatically verifying the correctness of termination certificates expressed in the CPF XML format as used in the termination competition. Termination certificates are currently translated and checked in Coq by using the CoLoR library. But a new standalone version is under development using Coq extraction mechanism (PhD subject of Kim-Quyen Ly).

Rainbow is distributed under the CeCILL license. It is currently developed by Frédéric Blanqui and Kim-Quyen Ly. See the website for more information.

6. New Results

6.1. Dedukti

The version 2.0 of the Dedukti system, developed by Ronan Saillard, has been released in July 2013. It is based on an improved version of the $\lambda\Pi$-calculus modulo where rewrite rules are explicitly added [31], and where the conditions for typing the rewrite rules are weakened.

This version is fully written in OCaml. It is smaller, far more efficient than the previous version, and permits to type-check much bigger files.

New features include a better reporting of errors, an interactive mode, an export functionality from Dedukti to the MMT format [53], and non-linear pattern matching.

6.2. Embeddings in the $\lambda\Pi$-calculus modulo

A new version of Coqine has been developed by Ali Assaf. This version is designed using a Coq plugin architecture, which allows for a smoother integration with Coq's code base and alleviates problems of maintainability that affected the previous version.

The implementation of Holide has been improved, by Ali Assaf. This improved version incorporates sharing at the level of terms and types. This optimization allows to reduce the type-checking time of the OpenTheory standard library from more than 30 minutes to less than 1 minute.

Catherine Dubois and Raphaël Cauderlier have studied a translation in the $\lambda\Pi$-calculus modulo of features coming from object oriented programming languages, such as inheritance and late binding. This compilation scheme has been applied to produce a new back-end for FoCaLize called Focalide, through a compilation to Dedukti. This translation can benefit from the flexibility of Dedukti to deal with more dynamic object-oriented languages than FoCaLiZe; they are currently working on generalizing this translation using ζ-calculus as a theoretical foundation for objects.

Resolution and superposition are proof-search methods that are used in state-of-the-art first-order automated theorem provers such as iProver, Vampire, E or SPASS. A shallow embedding of resolution and superposition proofs in the $\lambda\Pi$-calculus modulo has been proposed by Guillaume Burel, thus offering a way to check these proofs in a trusted setting, and to combine them with other proofs. This embedding has been implemented in particular as a backend of iProver Modulo, therefore allowing to check proofs found by iProver Modulo using Dedukti [20].

A shallow embedding in Dedukti of the tableaux proofs generated by Zenon modulo has been designed and implemented by Frédéric Gilbert [22], [23]. The embedding is based on a refined version of previous double-negation translations, introducing as less as possible double negations. This optimization has shown that more than half of the proofs found by Zenon modulo are not using the excluded-middle law, therefore being purely intuitionistic.

6.3. Automated Theorem Proving

Mélanie Jacquel (*Siemens*) and David Delahaye developed *Super Zenon* [5], a generalization of the extension of *Zenon* to superdeduction to handle any first order theory. To do so, they designed heuristics able to automatically transform axioms of a theory into rewrite rules. This new tool has been tested over the first order problems of the TPTP library and a significant increase has been observed. A first distribution of this tool (under GPL licence) is planned in the first months of 2014. In addition, an integration to the *Rodin* platform is also planned with the help of Laurent Voisin (*Systerel*). This integration should allow us to apply this tool in the context of *Event-B*.

Pierre Halmagrand, David Delahaye, Damien Doligez, and Olivier Hermant developed *Zenon Modulo* [22], [23], an extension of *Zenon* to Deduction modulo. Like *Super Zenon*, this new tool is able to deal with any first order theory and relies on an heuristic able to automatically transform axioms of a theory into rewrite rules. This tool has also been tested over the first order problems of the TPTP library and a similar increase of performance (compared to *Super Zenon*) has been observed. Frédéric Gilbert has developed a *Dedukti* backend for this extension, which is based on a double-negation transformation that allows us to transform classical proofs produced by *Zenon Modulo* into intuitionistic proofs in *Dedukti*. This tool is intended to be applied in the framework of the *BWare* project in order to automatically verify proof obligations coming from the modeling of industrial applications. To do so, the idea is to manually transform the *B* set theory into a theory modulo and provide it to *Zenon Modulo* in order to verify the proof obligations of the *BWare* project.

Guillaume Burel and Simon Cruanes have designed a method to scan sets of first-order clauses in order to detect the presence of instances of axiomatic theories (group structures, total orderings, etc.), even during a saturation process (so that theories that only become apparent during the proof search can be detected) [21]. To this end, they introduced the concept of *meta-prover*, a Datalog system that reasons over properties of the problem, and communicates with the saturation prover. This technique made some applications possible, such as the use of generic lemma and an equational redundancy criterion for some theories, and was implemented in Zipperposition.

Simon Cruanes has been working on superposition modulo linear arithmetic, using Zipperposition as a test bed. The focus is on problems with rational or integer arithmetic mixed with first-order reasoning, an area in which SMT solvers struggle. The work is still preliminary, but shows promising results.

Depending on the logic for finite structures, which is defined by Gilles Dowek and Ying Jiang (Beijing), Kailiang Ji has extended the use of proof search algorithms in Deduction modulo to automatically prove some graph properties, such as (un)reachability, which can be described by CTL formulas. A technical report about this has been given on Locali 2013 in Beijing.

Together with Tayssir Touili (University Paris Diderot) Hugo Macedo has shown how to advance the performance of the application of model checking techniques in the domain of malicious software detection. The work consisted in leveraging the reachability analysis used in the model checking of pushdown systems to infer malicious behavior patterns from known malware. From such new application a malware detection tool was prototyped and put to the test with instances of “in the wild” (real world) malicious software. This work was published in a large security venue and the details about the technique follow in [29].

Kim-Quyen Ly extended her formally-proved (in Coq) automated termination-certificate (for first-order rewrite systems) verifier Rainbow for dealing with certificates using arguments filtering [39] and other termination techniques.

6.4. Proof theory

The conservativity of the embedding of pure type systems in the $\lambda\Pi$-calculus modulo was proved by Ali Assaf. This result extends those of Cousineau and Dowek [46] and further justifies the use of the $\lambda\Pi$-calculus modulo as a logical framework. This embedding is the basis for the automated translation tools Holide and Coqine.

Frédéric Blanqui, Jean-Pierre Jouannaud (Univ. Paris 11) and Albert Rubio (Technical University of Catalonia) have developed a method aiming at carrying out termination proofs for higher-order calculi. CPO appears to be the ultimate improvement of the higher-order recursive path ordering (HORPO) [45] in the sense that this definition captures the essence of computability arguments *à la* Tait and Girard, therefore explaining the name of the improved ordering. It has been shown that CPO allows to consider higher-order rewrite rules in a simple type discipline with inductive types, that most of the guards present in the recursive calls of its core definition cannot be relaxed in any natural way without losing well-foundedness, and that the precedence on function symbols cannot be made more liberal anymore.

Frédéric Blanqui worked on the formalization in the Coq proof assistant of various definitions of the notion of α-equivalence on pure λ-terms. In particular, he formalized and formally proved equivalent the definitions of Church (1932), Curry and Feys (1958), Krivine (1993), and Gabbay and Pitts (1999). This work is freely available from the CoLoR library released on December 13th.

Alejandro Díaz-Caro and Gilles Dowek have introduced an extension of λ-calculus with pairs where isomorphic types are equated. Identifying some types requires to also identify some terms via an equivalence relation on terms, leading to an interesting calculus, which is related to several known non-deterministic and probabilistic calculi. A preliminary version of this work has been published on [24]. A complete version in simple types, with its proof of normalisation, is currently under review.

Together with Ying Jiang, Gilles Dowek has started to investigate the links between model-checking and proof-checking. This has materialized by an encoding of CTL for a finite model in predicate logic and by the definition of a proof-system for CTL.

Olivier Hermant has studied optimized versions of double-negation translations, that allow to switch between classical and intuitionistic logics. Such an algorithm has been implemented in Zenon's backend to Dedukti by Frédéric Gilbert. Gilles Dowek has given new version of Gödel's translation of classical logic into constructive logic. This translation is homomorphic, hence it can be seen as a mere definition of the classical connectives from the constructive ones.

6.5. Safety of aerospace systems

Pierre Néron has designed a method to transform straight line programs, such as those used in some aerospace systems into others that do not use some operations such as, square roots and divisions that cannot be performed exactly on decimal numbers. To this end he has defined a new notion of anti-unification, called *constrained anti-unification*, and a new anti-unification algorithm.

6.6. Models of Computation

Alejandro Díaz-Caro and Gilles Dowek have shown how to provide a structure of probability space to the set of execution traces on a non-confluent abstract rewrite system, by defining a variant of a Lebesgue measure on the space of traces. Then, they showed how to use this probability space to transform a non-deterministic calculus into a probabilistic one. As an example, they applied this technique to the previously introduced non-deterministic calculus. [25]

Ali Assaf and Alejandro Díaz-Caro, together with Simon Perdrix (Nancy), Christine Tasson (PPS) and Benoît Valiron (PPS) have determined the relationship between the algebraic λ-calculus, a fragment of the differential λ-calculus and the linear-algebraic λ-calculus, a candidate λ-calculus for quantum computation. Both calculi are algebraic: each one is equipped with an additive and a scalar-multiplicative structure, and their set of terms is closed under linear combinations. However, the two languages were built using different approaches: the former is a call-by-name language whereas the latter is call-by-value; the former considers algebraic equalities whereas the latter approaches them through rewrite rules. They have analysed how these different approaches relate to one another, proposing four canonical languages based on each of the possible choices: call-by-name versus call-by-value, algebraic equality versus algebraic rewriting. They have shown that the various languages simulate one another. Preliminary versions of this work where published in [47] and [41]. Now they are working on a journal version filling the gaps between these previous works.

Together with Pablo Arrighi (Grenoble) and Benoît Valiron (PPS), Alejandro Díaz-Caro has described a type system for the linear-algebraic lambda-calculus. The type system accounts for the linear-algebraic aspects of this extension of lambda-calculus: It is able to statically describe the linear combinations of terms that will be obtained when reducing the programs. This gives rise to an original type theory where types, in the same way as terms, can be superposed into linear combinations. They have proven that the resulting typed lambda-calculus is strongly normalising and features a weak subject reduction. In addition, they have shown how to naturally encode matrices and vectors in this typed calculus [34].

Gilles Dowek has investigated a new definition of the notion of a chaotic system that can be applied to discrete systems and that is compatible with the principle of a finite density of information.

The paper Call-by-value non-determinism in a linear logic type discipline by Alejandro Díaz-Caro, Giulio Manzonetto and Michele Pagani has been published [26].

The paper Universality in two dimensions of Gilles Dowek and Nachum Dershowitz has been published.

The paper Linear-algebraic lambda-calculus: higher-order, encodings and confluence of Pablo Arrighi and Gilles Dowek has been published.

The book *Lambda Calculus with Types* , written by Henk Barendregt, Wil Dekkers, Richard Statman, and 11 contributors, including Gilles Dowek, has been published.

6.7. Constraint solving

Catherine Dubois has extended the formally verified constraint solver (on finite domains) she has developed with Matthieu Carlier and Arnaud Gotlieb with a new local consistency property (bound-consistency).

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. This year we organized the first Locali workshop in Beijing.

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, coordinated by Pierre Valarcher, on models of computation.

7.2. International Initiatives

7.2.1. Informal International Partners

Deducteam and the KWARC research group (Jacobs University, Germany), led by Michael Kohlhase, have organized a common workshop in Paris on the 12 of April. This workshop has led to the two tools dk2MMT and MMT2dk, and another workshop is planned on the 2014 year. See the program at http://www.cri.ensmp.fr/people/hermant/deducteam/2013/kwarc-dedukti.html or the webpage of the seminars.

7.3. International Research Visitors

7.3.1. Visits of International Scientists

Hermann Haeusler, Bruno Bruno Lopes and Cecilia Englander, from the University PUC Rio have visited Deducteam.

Ying Jiang from the Institute of software of the Chinese Academy of Sciences has visited Deducteam.

7.3.2. Visits to International Teams

Gilles Dowek has visited the University PUC Rio and the Institute of software of the Chinese Academy of Sciences.

8. Dissemination

8.1. Scientific Animation

- Guillaume Burel has been a PC member of IWIL.
- David Delahaye has been a PC member of PxTP 2013. He is a member of the steering committee of Calculemus since 2010.
- Gilles Dowek has been a PC member of RTA, ICECCS, NFM and eMooc.
- Gilles Dowek is a member of the Cerna.

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

Bachelor: Ali Assaf, Les bases de la programmation et de l'algorithmique, 36 hours, third year, École Polytechnique, France

Bachelor: Ali Assaf, Algorithmique et programmation, 36 hours, third year, École Polytechnique, France

Bachelor: Alejandro Díaz-Caro, Mathématiques 1: Calcul et fonctions, 72 hours, 1st year, Université Paris X, France

Bachelor: Alejandro Díaz-Caro, Statistiques et probabilités, 6 hours, second year, Université Paris X, France

Bachelor: Alejandro Díaz-Caro, Mathématiques 2, 24 hours, first year, Université Paris X, France

Bachelor: Alejandro Díaz-Caro, Méthodologie de la mesure en sciences humaines, 48 hours, first year, Université Paris X, France

Bachelor: Ronan Saillard, Programmation Orientée Objet en Java, 27 hours ,Telecom ParisTech, France

Bachelor: Guillaume Burel, Programmation avancée, 25,5 hours, third year, ENSIIE, France

Bachelor: Guillaume Burel, Logique, 10,5 hours, third year, ENSIIE, France

Bachelor: Guillaume Burel, Projet informatique, 22,75 hours, third year, ENSIIE, France

Master: Guillaume Burel, Systèmes et langages formels, 21 hours, first year, ENSIIE, France

Master: Guillaume Burel, Compilation, 36,75 hours, first year, ENSIIE, France

Master: Guillaume Burel, Sémantique des langages de programmation, 21hours, second year, ENSIIE, France

Master: Pierre Halmagrand, Sureté fonctionnelle, 12,5 hours, second year, CNAM Saint-Denis, France

Master: Olivier Hermant, Méthodes formelles, 21 hours, second year, ISEP, France

Master: Olivier Hermant, Algorithmique, 9 hours, second year, ISEP, France

Master: Olivier Hermant, Complétude et élimination des coupures, 1,5 hours, second year, Université Paris Diderot, France

Master: Gilles Dowek, Fondement des systèmes de preuves, 27 hours, second year, MPRI.

Guillaume Burel is in charge of the 4th, 5th, and 6th semesters of the engineering degree at ENSIIE.

David Delahaye has taught at *Cnam* courses in the following topics: Algorithmics, Object-Oriented Programming Languages, Safety and Security, Formal Proofs and Automated Deduction.

8.2.2. Supervision

PhD in progress : Ali Assaf, Interoperabilty of proof systems, September 2012, Gilles Dowek and Guillaume Burel

PhD in progress: Raphaël Cauderlier, Mécanismes orientés objets pour le raffinement de données et d'algorithmes dans les systèmes de preuve, September 2013, Catherine Dubois

PhD in progress : Simon Cruanes, Automated reasoning modulo theories, August 2012, Gilles Dowek and Guillaume Burel

PhD in progress : Kailiang Ji, Model checking and automated theorem proving, September 2012, Gilles Dowek

PhD in progress: Kim-Quyen Ly, automated formal verification of termination certificates, October 2011, Frédéric Blanqui

PhD in progress: Pierre Halmagrand, Déduction automatique modulo, November 2013, David Delahaye and Olivier Hermant and Damien Doligez

PhD in progress: Vivien Maisonneuve, Préservation de preuve de système lors de la compilation sur micro-contrôleur, October 2011, François Irigoin and Olivier Hermant

PhD: Pierre Néron: A Quest for Exactness: Program Transformation for Reliable Real Numbers, December 2013, Gilles Dowek.

PhD in progress: Ronan Saillard, Dedukti: un vérificateur de preuves universel, October 2012, Pierre Jouvelot and Olivier Hermant

- David Delahaye, Olivier Hermant, and Damien Doligez have supervised the Master internship of Pierre Halmagrand.
- Catherine Dubois has supervised the undergrad internship of Frédéric Lang.

8.2.3. PhD and Habilitation juries

- Catherine Dubois has been a member of the PhD juries of Mounira Kezadri and Asma Tafat.
- Gilles Dowek has been a member of the Phd juries of Jianhua Gao, Maël Pégny, Alberto Naibo, and to the HDR jury of Benjamin Nguyen.

8.3. Popularization

- **Seminars in international workshops without peer review**
  - Alejandro Díaz-Caro. Identifying isomorphic propositions. In *First Workshop of ANR-NSFC project LOCALI*. Beijing, China. November 4–6.
  - Pierre Halmagrand. Proof compression and certification in Zenon Modulo. In *Third Workshop of the Amadeus Project for Proof Compression*. Nancy, France. September 16.
  - Gilles Dowek has participated to the workshop Locali in Beijing where he has given a talk.
- **Seminars in national workshops without peer review**
  - Alejandro Díaz-Caro. Identifying isomorphic propositions. In *Journées LAC*. Créteil, France. November 28–29.
  - Alejandro Díaz-Caro. Vectorial types, non-determinism and probabilistic systems: Towards a computational quantum logic. In *Quantum Computing in Nancy*. Nancy, France. March 21.
  - Gilles Dowek has participated to the workshop Rochebrune 2013 *La preuve et ses moyens*, where he has given a talk.

  - Gilles Dowek has given several talks on Computer Science in Education, at the congrès de la SIF, the workshop EPI in Nancy, the workshop UPS in Luminy, the journée ISN in Orsay, the journée Pascaline in Paris, etc.
  - Gilles Dowek has given several popular science talks: in Paris for the young kangaroos, in Athens, and in the Lycée Raoul Folleraux.
- **Other seminars**
  - Alejandro Díaz-Caro. Hacia una lógica computacional cuántica. FCEIA, Universidad Nacional de Rosario. Rosario, Argentine. August 9.
  - Alejandro Díaz-Caro. Vectorial types, non-determinism and probabilistic systems: Towards a quantum computational logic. PPS, Université Paris-Diderot. Paris, France. May 7.
  - Alejandro Díaz-Caro. Vectorial types, non-determinism and probabilistic systems: Towards a quantum computational logic. LIAFA, Université Paris-Diderot. Paris, France. April 16.
  - Alejandro Díaz-Caro. Non determinism (and probabilities) through type isomorphism. LIP, École Normale Supérieure. Lyon, France. February 21.
  - Alejandro Díaz-Caro. Quantum computing, non-determinism, probabilistic systems...and the logic behind. Modal'X. Université Paris X. Nanterre, France. January 31.
  - Pierre Halmagrand. Zenon Modulo: When Achilles outruns the tortoise using Deduction modulo. Mines ParisTech laboratory CRI. Fontainebleau, France. October 30.

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] D. DELAHAYE, M. JACQUEL. *Recovering Intuition from Automated Formal Proofs using Tableaux with Superdeduction*, in "Electronic Journal of Mathematics and Technology (eJMT)", February 2013, vol. 7, n^{o} 2

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

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

[8] 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

[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, To appear

Publications of the year

Doctoral Dissertations and Habilitation Theses

[11] P. NERON. , *A Quest for Exactness: Program Transformation for Reliable Real Numbers*, Ecole Polytechnique X, October 2013, http://hal.inria.fr/tel-00924379

Articles in International Peer-Reviewed Journals

[12] P. ARRIGHI, G. DOWEK. *Causal graph dynamics*, in "Information and Computation", 2013, vol. 223, pp. 78-93 [*DOI :* 10.1016/J.IC.2012.10.019], http://hal.inria.fr/hal-00944459

[13] P. ARRIGHI, G. DOWEK. *Lineal: A linear-algebraic lambda-calculus*, in "Logical Methods in Computer Science", 2013, http://hal.inria.fr/hal-00919625

[14] N. DERSHOWITZ, G. DOWEK. *Universality in two dimensions*, in "Journal of Logic and Computation", 2013 [*DOI :* 10.1093/LOGCOM/EXT022], http://hal.inria.fr/hal-00919604

[15] H. MACEDO, J. OLIVEIRA. *Typing linear algebra: A biproduct-oriented approach*, in "Science of Computer Programming", 2013, vol. 78, n° 11, pp. 2160-2191 [*DOI :* 10.1016/J.SCICO.2012.07.012], http://hal.inria.fr/hal-00919866

[16] P. NERON. *A Formal Proof of Square Root and Division Elimination in Embedded Programs*, in "Journal of Formalized Reasoning", December 2013, vol. 6, n° 1, pp. 89-111, http://hal.inria.fr/hal-00924367

Invited Conferences

[17] G. DOWEK. *Real numbers, chaos, and the principle of a bounded density of information*, in "CSR 2013 - 8th International Computer Science Symposium in Russia", Ekaterinburg, Russian Federation, A. A. BULATOV, A. M. SHUR (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 7913, pp. 347-353 [*DOI :* 10.1007/978-3-642-38536-0_30], http://hal.inria.fr/hal-00919543

International Conferences with Proceedings

[18] L. ALLALI, O. HERMANT. *Semantic A-translation and Super-consistency entail Classical Cut Elimination*, in "LPAR 19 - 19th Conference on Logic for Programming, Artificial Intelligence, and Reasoning - 2013", Stellenbosch, South Africa, K. MCMILLAN, A. MIDDELDORP, A. VORONKOV (editors), Lecture Notes in Computer Science, Springer, December 2013, vol. 8312, pp. 407-422 [*DOI :* 10.1007/978-3-642-45221-5_28], http://hal.inria.fr/hal-00923915

[19] M. BOUDARD, O. HERMANT. *Polarizing Double Negation Translations*, in "LPAR", Stellenbosch, South Africa, K. MCMILLAN, A. MIDDELDORP, A. VORONKOV (editors), LNCS ARCoSS, Springer, 2013, vol. 8312, pp. 182-197, http://hal.inria.fr/hal-00920224

[20] G. BUREL. *A Shallow Embedding of Resolution and Superposition Proofs into the λΠ-Calculus Modulo*, in "PxTP - Third International Workshop on Proof Exchange for Theorem Proving - 2013", Lake Placid, United States, J. C. BLANCHETTE, J. URBAN (editors), EasyChair, June 2013, vol. 14, pp. 43-57, http://hal.inria.fr/hal-00921513

[21] G. BUREL, S. CRUANES. *Detection of First Order Axiomatic Theories*, in "FroCoS - 9th International Symposium on Frontiers of Combining Systems - 2013", Nancy, France, P. FONTAINE, C. RINGEISSEN, R. A. SCHMIDT (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 8152, pp. 229-244 [*DOI :* 10.1007/978-3-642-40885-4_16], http://hal.inria.fr/hal-00919759

[22] 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, http://hal.inria.fr/hal-00909688

[23] 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), LNCS, Springer, December 2013, vol. 8312, pp. 274-290 [*DOI :* 10.1007/978-3-642-45221-5_20], http://hal.inria.fr/hal-00909784

[24] A. DÍAZ-CARO, G. DOWEK. *Non determinism through type isomorphism*, in "LSFA - 7th Workshop on Logical and Semantic Frameworks with Applications - 2012", Rio de Janeiro, Brazil, D. KESNER, P. VIANA (editors), Electronic Proceedings in Theoretical Computer Science, Open Publishing Association, March 2013, vol. 113, pp. 137-144 [*DOI :* 10.4204/EPTCS.113.13], http://hal.inria.fr/hal-00925001

[25] A. DÍAZ-CARO, G. DOWEK. *The probability of non-confluent systems*, in "DCM - 9th International Workshop on Developments in Computational Models - 2013", Buenos Aires, Argentina, M. A. RINCÓN, E. BONELLI, I. MACKIE (editors), 2013, To appear in EPTCS, http://hal.inria.fr/hal-00919546

[26] A. DÍAZ-CARO, G. MANZONETTO, M. PAGANI. *Call-by-value non-determinism in a linear logic type discipline*, in "LFCS - Logical Foundations of Computer Science - 2013", San Diego, CA, United States, S. ARTEMOV, A. NERODE (editors), Lecture Notes in Computer Science, Springer, January 2013, vol. 7734, pp. 164-178 [*DOI :* 10.1007/978-3-642-35722-0_12], http://hal.inria.fr/hal-00919463

[27] T. G. LE, D. FEDOSOV, O. HERMANT, M. MANCENY, R. PAWLAK, R. RIOBOO. *Programming Robots With Events*, in "IESS 2013 - 4th IFIP TC 10 International Embedded Systems Symposium", Paderborn, Germany, G. SCHIRNER, M. GÖTZ, A. RETTBERG, M. C. ZANELLA, F. J. RAMMIG (editors), IFIP Advances in Information and Communication Technology, Springer, 2013, vol. 403, pp. 14-25 [*DOI :* 10.1007/978-3-642-38853-8_2], http://hal.inria.fr/hal-00924489

[28] T. G. LE, O. HERMANT, M. MANCENY, R. PAWLAK, R. RIOBOO. *Using Event-Based Style for Developing M2M Applications*, in "GPC - 8th International Conference on Grid and Pervasive Computing - 2013", Seoul, Korea, Republic Of, J. PARK, H. ARABNIA, C. KIM, W. SHI, J.-M. GIL (editors), Lecture Notes in Computer

Science, Springer, 2013, vol. 7861, pp. 348-357 [*DOI :* 10.1007/978-3-642-38027-3_37], http://hal.inria.fr/hal-00924491

[29] H. MACEDO, T. TOUILI. *Mining Malware Specifications through Static Reachability Analysis*, in "ESORICS - European Symposium on Research in Computer Security", Egham, United Kingdom, J. CRAMPTON, S. JAJODIA, K. MAYES (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 8134, pp. 517-535 [*DOI :* 10.1007/978-3-642-40203-6_29], http://hal.inria.fr/hal-00919782

[30] P. NERON. *Square root and division elimination in PVS*, in "ITP - 4th Conference on Interactive Theorem Proving", Rennes, France, S. BLAZY, C. PAULIN-MOHRING, D. PICHARDIE (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 7998, pp. 457-462 [*DOI :* 10.1007/978-3-642-39634-2_33], http://hal.inria.fr/hal-00924359

[31] R. SAILLARD. *Towards explicit rewrite rules in the λΠ-calculus modulo*, in "IWIL - 10th International Workshop on the Implementation of Logics", Stellenbosch, South Africa, December 2013, http://hal.inria.fr/hal-00921340

Scientific Books (or Scientific Book chapters)

[32] J. COURANT, M. DE FALCO, S. GONNORD, J.-C. FILLIÂTRE, S. CONCHON, G. DOWEK, B. WACK. , *Informatique pour tous en classes préparatoires aux grandes écoles : Manuel d'algorithmique et programmation structurée avec Python*, Eyrolles, 2013, 408 p. , http://hal.inria.fr/hal-00880268

Research Reports

[33] A. ROUSSEAU, A. DARNAUD, B. GOGLIN, C. ACHARIAN, C. LEININGER, C. GODIN, C. HOLIK, C. KIRCHNER, D. RIVES, E. DARQUIE, E. KERRIEN, F. NEYRET, F. MASSEGLIA, F. DUFOUR, G. BERRY, G. DOWEK, H. ROBAK, H. XYPAS, I. ILLINA, I. GNAEDIG, J. JONGWANE, J. EHREL, L. VIENNOT, L. GUION, L. CALDERAN, L. KOVACIC, M. COLLIN, M.-A. ENARD, M.-H. COMTE, M. QUINSON, M. OLIVI, M. GIRAUD, M. DORÉMUS, M. OGOUCHI, M. DROIN, N. LACAUX, N. ROUGIER, N. ROUSSEL, P. GUITTON, P. PETERLONGO, R.-M. CORNUS, S. VANDERMEERSCH, S. MAHEO, S. LEFEBVRE, S. BOLDO, T. VIÉVILLE, V. POIREL, A. CHABREUIL, A. FISCHER, C. FARGE, C. VADEL, I. ASTIC, J.-P. DUMONT, L. FÉJOZ, P. RAMBERT, P. PARADINAS, S. DE QUATREBARBES, S. LAURENT. , *Médiation Scientifique : une facette de nos métiers de la recherche*, March 2013, 34 p. , http://hal.inria.fr/hal-00804915

Other Publications

[34] P. ARRIGHI, A. DÍAZ-CARO, B. VALIRON. , *The Vectorial Lambda-Calculus*, 2013, Under review, http://hal.inria.fr/hal-00921087

[35] G. DOWEK. , *On the definition of the classical connectives and quantifiers*, December 2013, http://hal.inria.fr/hal-00919437

[36] G. DOWEK, Y. JIANG. , *A Logical Approach to CTL*, January 2014, http://hal.inria.fr/hal-00919467

[37] G. DOWEK, Y. JIANG. , *Axiomatizing truth in a finite model*, January 2014, http://hal.inria.fr/hal-00919469

[38] P. NERON. *Elimination des racines et divisions pour du code embarqué*, in "Journées du GDR-GPL", Nancy, France, April 2013, Journées du GDR-GPL, http://hal.inria.fr/hal-00924394

References in notes

[39] T. ARTS, J. GIESL. *Termination of Term Rewriting Using Dependency Pairs*, in "Theoretical Computer Science", 2000, vol. 236, pp. 133-178

[40] A. ASPERTI, W. RICCIOTTI, C. SACERDOTI COEN, E. TASSI. *A compact kernel for the calculus of inductive constructions*, in "Sadhana", 2009, vol. 34, n^{o} 1, pp. 71-144, http://dx.doi.org/10.1007/s12046-009-0003-3

[41] A. ASSAF, S. PERDRIX. *Completeness of algebraic CPS simulations*, in "DCM'11", EPTCS, 2012, vol. 88, pp. 16–27

[42] L. BENTHEM JUTTING, J. MCKINNA, R. POLLACK. *Checking algorithms for Pure Type Systems*, in "Types for Proofs and Programs", H. BARENDREGT, T. NIPKOW (editors), Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1994, vol. 806, pp. 19-61

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

[44] F. BLANQUI. , *Terminaison des systèmes de réécriture d'ordre supérieur basée sur la notion de clôture de calculabilité*, Université Paris-Diderot - Paris VII, July 2012, HDR, http://tel.archives-ouvertes.fr/tel-00724233

[45] F. BLANQUI, J.-P. JOUANNAUD, A. RUBIO. *The Computability Path Ordering: the End of a Quest*, in "Proceedings of the 22nd International Conference on Computer Science Logic, Lecture Notes in Computer Science 5213", 2008, Invited paper

[46] 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

[47] A. DÍAZ-CARO, S. PERDRIX, C. TASSON, B. VALIRON. *Equivalence of Algebraic λ-calculi*, in "Informal proceedings of HOR-2010", Edinburgh, UK, July 14, 2010, pp. 6–11

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

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

[50] 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

[51] J. HURD. *The OpenTheory Standard Theory Library*, in "NASA Formal Methods", M. BOBARU, K. HAVELUND, G. HOLZMANN, R. JOSHI (editors), Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2011, vol. 6617, pp. 177-191, http://dx.doi.org/10.1007/978-3-642-20398-5_14

[52] 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

[53] 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