Activity Report 2012

Team DEDUCTEAM

Deduction modulo, interopérabilité et démonstration automatique

RESEARCH CENTER
Paris - Rocquencourt

THEME
Programs, Verification and Proofs

Table of contents

Team DEDUCTEAM

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

Creation of the Team: December 01, 2011 , *Updated into Exploratory Action:* January 01, 2013 .

1. Members

Research Scientists
Gilles Dowek [Team leader, Senior Researcher Inria, HdR]
Catherine Dubois [Senior Researcher Inria, HdR]
Olivier Hermant [Mines ParisTech]

Faculty Members
Guillaume Burel [Associate Professor, ENSIIE/Cedric]
David Delahaye [Associate Professor, CNAM/Cedric, HdR]
Alejandro Díaz-Caro [Ater, Paris 10]

PhD Students
Ali Assaf [École polytechnique]
Simon Cruanes [École polytechnique]
Kailiang Ji [Paris Diderot]
Pierre Néron [École polytechnique]
Ronan Saillard [Mines ParisTech]

Visiting Scientists
Nachum Dershowitz [Tel Aviv]
Cecilia Englander [Puc-Rio]

Administrative Assistant
Hélène Milome

Others
Raphaël Bost [Intern, École polytechnique]
Quentin Carbonneaux [Intern, MPRI]
Raphaël Cauderlier [Intern, École Normale Supérieure de Cachan]

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 [6], [7] 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 [53], where the equivalence relation is definitional equality, Prawitz' extended natural deduction [54], 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 [6], [9]—Resolution modulo tableaux modulo, etc.—that could be applied to any theory formulated in Deduction modulo, to general proof normalization and cut elimination results [7], to the definitions of models taking the difference between reasoning and computation into account, and to the definition of specific theories—simple type theory [5], arithmetic, some versions of set theory, etc.—as purely computational theories.

2.3. From proof-checking to Interoperability

A new turn 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 [3]. 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 [1], that could be used to verify proofs coming from different proof systems, such as Coq [47], 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 and HOLiDe, that permit to translate proofs from Coq and HOL to Dedukti.

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* [27].

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.

2.4. 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 [4] 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 [2], 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 exploited for tableaux methods leading to the systems ZenonB and super-Zenon.

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.

2.5. 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.

3. Application Domains

3.1. 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 Munoz' 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.

3.2. B-set theory

The B method allows the user to develop software correct by construction, going from abstract models to implementations via refinement. During the development process, proof obligations are generated. The formalism underlying B is based on predicate logic and B-set theory. Atelier B that supports the B method provides interactive and automatic provers. To increase automation the user may add proof rules which, if not correct, may corrupt the process. Siemens has developed a tool chain to verify such added proof rules. In particular we have to verify that any proof rule derives from B logic. This step has to be as automatic as possible. Furthermore confidence in these verification proofs is required. A first attempt using the first order prover Zenon allowed the verification of a large number of proof rules [10]. To go further we have experimented techniques such as super deduction and deduction modulo. B-set theory is an interesting benchmark for the tools developed by Deducteam since this theory contains numerous operators and predicates defined by equations or rewrite rules.

4. Software

4.1. 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 is designed to be versatile: it must be efficient on proofs that contain many computations—such as proofs by reflection—as well as proofs that do not contain any–such as proofs coming from HOL. These constraints has led us to adopt a Just-In-Time compilation architecture. And instead of designing our own JIT compiler, we have chosen to reuse the cutting-edge LuaJIT compiler. This technological choice, namely devolving the type-checking to Lua, makes Dedukti a proof-checker generator.

This has allowed the introduction of many optimizations: a normalization by evaluation strategy, a higher-order abstract syntax representation of terms and a context-free, bidirectional type-checking algorithm [22].

Dedukti has been developed by Mathieu Boespflug, Olivier Hermant, Quentin Carbonneaux, and Ronan Saillard.

4.2. CoqInE and HOLiDe

Dedukti comes with two companion tools: HOLiDe, an embedding of HOL proofs through the OpenTheory format [51], and CoqInE, an embedding of Coq proofs. Almost all the standard library of HOL and a significant part of that of Coq are checked by Dedukti.

CoqInE now supports the following features of Coq: the raw Calculus of Constructions, inductive types, and fixpoint definitions. It is now able to translate more than 80% of the standard library of Coq [21]. Ongoing work focuses on modules and functors, and on universes.

CoqInE has been developed by Mathieu Boespflug, Guillaume Burel, and Ali Assaf.

HOLiDe supports all the features of HOL, including 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.

4.3. 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.

It is developed by Guillaume Burel.

These four systems are available on the website of the team.

5. New Results

5.1. Dedukti

Together with Mathieu Boespflug (McGill University), Quentin Carbonneaux and Ronan Saillard have developed a new version of the front-end of Dedukti, written in OCaml, replacing an inefficient previous version, as well as a new version of the back-end using the Lua Just-In-Time compiler.

Ronan Saillard has internalized the Lua back-end of Dedukti, so that it is no longer necessary to explicitly call it when using Dedukti.

Ronan Saillard has extended the input language of Dedukti to allow the user to declare dependencies between modules, to write definition or to explicitly require to type-check a term.

Ronan Saillard has added a new feature to Dedukti to make opaque definitions. As with usual definitions, the proof term of an opaque definition is type-checked, but it is then immediately forgotten in order to decrease memory consumption.

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

Ali Assaf has designed an embedding of the HOL logic in the $\lambda\Pi$-calculus modulo and implemented it in the HOLiDe system [40].

Together with Mathieu Boespflug, Ali Assaf and Guillaume Burel have developed an embedding of the Calculus of Inductive Constructions with universes in the $\lambda\Pi$-calculus modulo and Ali Assaf is currently implementing it in a new version of the CoqInE system.

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 [8], through a compilation to Dedukti. This new back-end is expected to be lighter than the present one producing Coq code and also to be able to combine local and external proofs coming from different proof environments [44]. They are currently working on a translation of the full FoCaLize language—not restricted to its object oriented features—and on a proof of its correctness with respect to the existing FoCaLize semantics.

5.3. Automated Theorem Proving

Guillaume Burel has shown that presenting theories by means of rewriting rules in Deduction modulo leads to more efficient proof search methods than using axioms, provided the rewriting system enjoys a proof theoretical property, namely cut admissibility.

He has been investigating which theories can be encoded as rewriting systems admitting cuts. Surprisingly, it turned out that any consistent theory in predicate logic can. This has been shown by studying the links between the set-of-support strategy of the Resolution method and the extension of the method based on Deduction modulo. He has also shown how to reduce the size of the corresponding rewriting systems [42].

Guillaume Burel has also studied how to improve the confidence in iProver Modulo. When it finds a resolution proof, it is now able to produce a proof that can be checked by Dedukti. The encoding of Resolution proofs in the $\lambda\Pi$-calculus modulo that is used is shallow, making more plausible the long-term goal of interoperability of provers, both interactive and automated, through Dedukti.

Simon Cruanes has explored several ideas for combining the Superposition calculus—one of the most powerful calculi for automated reasoning within first-order logic with equality—with Deduction modulo. Combining the term rewriting system for a theory in Deduction modulo with the ordered rewriting on which Superposition is based on proved to be difficult, yielding incomplete calculi; in most cases it boils down to the fact that the combination of confluent terminating term rewriting systems is in general neither terminating nor confluent. In order to experiment quickly ideas by implementing them, he has written a Superposition-based prover in OCaml, with some special features—automatic ordering of rewrite rules in the input, non-clausal calculus to be able to use equivalence relations as rewrite rules. The prover is 8,000 lines of code and is designed to be flexible and modular, but still has decent performance and can prove some non-trivial theorems.

Together with Mélanie Jacquel (Cedric), David Delahaye and Catherine Dubois have investigated Zenon for verifying proof rules added to help the automation in the provers of Atelier B. They have augmented Zenon with specific rules for dealing with set operations and predicates, obtained by applying super deduction—a variant of Deduction modulo [33].

5.4. Proof theory

We believe that our work on proof-checking and automated theorem proving cannot be separated from a more theoretical research on proof theory.

Together with Denis Cousineau, Gilles Dowek and Olivier Hermant have related semantic criteria for proof normalization and admissibility of the cut rule in Deduction modulo [17], [26].

Gilles Dowek has proposed a new way to define classical connectives in a constructive framework [46].

Together with Murdoch J. Gabbay (Heriot Watt), Gilles Dowek has proposed a new nominal logic that handles binders in terms [16] and a new semantics for predicate logic [29].

During her visit in the team, Cecilia Englander has studied the correspondence between natural deduction and sequent calculus.

Together with Ying Jiang (Beijing), Gilles Dowek has defined a logic for finite structures. Kailiang Ji is currently investigating the use of proof search algorithms in Deduction modulo to automatically prove theorems in this theory.

5.5. Safety of aerospace systems

Together with Anthony Narkawicz (Nasa-Langley) and César Muñoz (Nasa-Langley), Gilles Dowek has designed a prevention bands algorithm, that is an algorithm that computes and displays to the pilot of an aircraft, a sequence of safe and unsafe intervals on ground speed, heading or vertical speed and they have proved this algorithm correct in the PVS system [18].

This algorithm computes with real numbers, but its implementation computes with floating point numbers. Moreover this algorithm is numerically unstable as it uses comparisons of numbers, computed with square root and division operations. This has led Pierre Néron to design a program transformation algorithm to eliminate square roots and divisions in straight-line programs. This way computation can be made exact.

Together with César Muñoz, Pierre Néron has completed this year the design of this program transformation algorithm and he has proved, in the PVS system, its termination and correctness: preservation of semantics and absence of square roots and divisions in the produced program [35].

Together with César Muñoz, Pierre Néron has also implemented this transformation algorithm as a PVS automatic proof strategy, that allows a wider range of expressions, using a deep embedding of PVS in PVS itself.

Pierre Néron and Raphaël Bost have proposed an optimization of one aspect of that algorithm: the definition of a common template for arithmetic expression.

5.6. Constraint Solving

Catherine Dubois has developed in collaboration with Matthieu Carlier and Arnaud Gotlieb (Oslo) a formally verified constraint finite domain solver. It focuses on arc-consistency and has been developed with Coq [24].

5.7. Models of Computation

Together with Pablo Arrighi (Grenoble), Gilles Dowek has reformulated Gandy's proof of the physical Church-Thesis in the quantum case [11]. Gilles Dowek has proposed the idea that the Galileo thesis could be seen as a consequence of the physical Church-Turing thesis and therefore as a consequence of Gandy's principles [15]. Gilles Dowek has proposed a definition of a notion of non deterministic computation over the real numbers [14] that could be used as a language to describe continuous non deterministic physical phenomena. All this work has then been presented in a tutorial at the conference *Language and Automata Theory and Applications* [28].

Together with Pablo Arrighi, Gilles Dowek has investigated further the principle of a finite density of information [38] and in particular the impact of this definition on the notion of a chaotic dynamical system [37].

Together with Pablo Arrighi, Gilles Dowek has investigated a generalization of the notion of cellular automaton where the principle of a bounded density of information is formulated independently of the geometry of space. This led to the notion of a Causal graph dynamic [12].

Nachum Dershowitz and Gilles Dowek have shown that extending Turing machines with a two-dimensional tape, made this formalism usable in practice to implement classical algorithms [45].

Alejandro Díaz-Caro and Gilles Dowek have proposed to take a fresh look at non deterministic λ-calculi—such as quantum λ-calculi—and derive non determinism from type isomorphism [30].

Together with Giulio Manzonetto (Paris 13) and Michele Pagani (Paris 13), Alejandro Díaz-Caro has considered an extension of the call-by-value λ-calculus with a may-convergent non-deterministic choice and a must-convergent parallel composition, endowed with a type system. They have proved that a term is typable if and only if it is converging, and that its typing tree carries enough information to give a bound on the length of its lazy call-by-value reduction. Moreover, when the typing tree is minimal, such a bound becomes the exact length of the reduction [31].

Together with Barbara Petit (Sardes), Alejandro Díaz-Caro has considered the non-deterministic extension of the call-by-value lambda calculus, which corresponds to the additive fragment of the linear-algebraic lambda-calculus. They have defined a fine-grained type system, capturing the right linearity present in such formalisms. After proving the subject reduction and the strong normalisation properties, they have proposed a translation of this calculus into the System F with pairs, which corresponds to a non linear fragment of linear logic. The translation provides a deeper understanding of the linearity in this setting [32].

Together with Pablo Arrighi, Barbara Petit, Pablo Burias (Rosario), Mauro Jaskelioff (Rosario), and Benoît Valiron (Penn), Alejandro Díaz-Caro has studied possible typing systems for the full linear-algebraic λ-calculus in which the non-deterministic calculus can be seen as a particular case. They have proposed a type system that keeps track of "the amount of a type" that is present in each term [13]. As an example of its use, they have shown that it can serve as a guarantee that the normal form of a term is barycentric, that is that its scalars are summing to one. They also proposed a type system similar to the one presented in [32], but for the full calculus, ensuring confluence and convergence [23]. Finally, they provided a full type system that is able to statically describe the linear combinations of terms resulting from the reduction of programs, also ensuring convergence [19].

6. Partnerships and Cooperations

6.1. National Initiatives

6.1.1. ANR Locali

We are coordinators of the ANR-NFSC contract Locali with the Chinese Academy of Sciences. This year we mostly developed in proof in a finite structure project of this contract.

6.1.2. ANR BWare

We are members of the ANR Beware which started on last September (David Delahaye is the national leader). The objective is to provide a proof platform for B proof obligations. We are in particular involved in the introduction of Deduction modulo in the automated proved tableaux-based Zenon and also in the combination of Deduction modulo and superposition.

6.1.3. ANR Tarmac

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

6.2. International Research Visitors

6.2.1. Visits of International Scientists

Nachum Dershowitz (Tel Aviv) has been visiting our group for three months.

Cecilia Englander (Puc-Rio) has been visiting our group for four months.

6.2.2. Visits to International Teams

Pierre Néron has been visiting César Muñoz group in Nasa-Langley for three months.

7. Dissemination

7.1. Scientific Animation

Gilles Dowek has been a PC member of LICS, CSL, TAMC, and ICECCS. He is a member of the Scientific Board of the *Société Informatique de France*. He is a member of the *Commission de réflexion sur l'éthique de la recherche en sciences et technologies du numérique d'Allistene*. He is Deputy scientific director of Inria.

Catherine Dubois has been a PC member of TAP and AFADl. She has been the chair of the TAP steering committee since May 2012.

7.2. Teaching - Supervision - Juries

7.2.1. Teaching activities

Olivier Hermant has taught at ISEP four engineering courses: Preparation to the ACM programming contest, Data bases, Formal methods, and Algorithmics.

Guillaume Burel has taught at ENSIIE five courses: Logic, Formal languages, Compilation, and Semantics of programming languages.

Alejandro Díaz-Caro has been teaching assistant at Paris 10 for four courses: in Mathematics (in two different degrees) and in Methodology of measure in social sciences.

Pierre Néron has been a teaching assistant in Polytech' Jussieu for two courses: Project in C and Introduction to networks.

Gilles Dowek has taught in MPRI a course: Foundations of proof systems. He has coordinated a textbook for high school students [39]. He is a member of the Scientific board of *La main à la pâte*.

7.2.2. Ph.D. supervision

Olivier Hermant supervises the theses of Ronan Saillard, Giang Le Truong, and Vivien Maisonneuve.

Catherine Dubois and David Delahaye supervise the thesis of Mélanie Jacquel and Pierre Nicolas Tollitte.

Catherine Dubois and Francois Pessaux (ENSTA) supervise the thesis of Vincent Benayoun.

Catherine Dubois supervises the research internship of Raphaël Cauderlier.

Guillaume Burel and Gilles Dowek supervise the theses of Ali Assaf and Simon Cruanes.

Gilles Dowek supervises the thesis of Pierre Néron.

7.2.3. Supervision of Masters internship

Catherine Dubois and Olivier Hermant have supervises the Masters internship of Raphaël Cauderlier.

Olivier Hermant and Mathieu Boespflug have supervised the Masters internship of Quentin Carbonneaux.

Guillaume Burel and Gilles Dowek have supervised the Masters internship of Ali Assaf.

Gilles Dowek and Pierre Néron have supervised the Masters internship of Raphaël Bost.

7.2.4. Juries

Gilles Dowek has been a member of the Jury of Vincent Demange. he has been a member of the Jury of the habilitations of Frédéric Blanqui and Sylvain Conchon.

Catherine Dubois has been a member of the jury of Maximiliano Cristia and a member of the jury of Ophaina Taofifenua.

7.3. Popularization

Gilles Dowek has given a talk at the Centre d'Alembert, at the Lycée Jacquard in Paris and at the Lycée Montaigne in Bordeaux.

8. Bibliography

Major publications by the team in recent years

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

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

[3] 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, p. 102-117.

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

[5] G. DOWEK, T. HARDIN, C. KIRCHNER. *HOL-lambda-sigma: an intentional first-order expression of higher-order logic*, in "Mathematical Structures in Computer Science", 2001, vol. 11, p. 1-25.

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

[7] G. DOWEK, B. WERNER. *Proof normalization modulo*, in "The Journal of Symbolic Logic", 2003, vol. 68, n^{o} 4, p. 1289-1316.

[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, p. 33-48.

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

[10] M. JACQUEL, K. BERKANI, D. DELAHAYE, C. DUBOIS. *Verifying B Proof Rules Using Deep Embedding and Automated Theorem Proving*, in "Software Engineering and Formal Methods - 9th International Conference, SEFM 2011, Montevideo, Uruguay, November 14-18, 2011. Proceedings", G. BARTHE, A. PARDO, G. SCHNEIDER (editors), Lecture Notes in Computer Science, Springer, 2011, vol. 7041, p. 253-268.

Publications of the year

Articles in International Peer-Reviewed Journals

[11] P. ARRIGHI, G. DOWEK. *The physical Church-Turing thesis and the principles of quantum theory*, in "International Journal of Foundations of Computer Science", 2012, vol. 23, n^{o} 5, 10.1142/S0129054112500153.

[12] P. ARRIGHI, G. DOWEK. *Causal Graph Dynamics*, in "Information and Computation", 2013, vol. 223, p. 78-93, 10.1016/j.ic.2012.10.019.

[13] P. ARRIGHI, A. DÍAZ-CARO. *A System F accounting for scalars*, in "Logical Methods in Computer Science", 2012, vol. 8, n^{o} 1:11.

[14] G. DOWEK. *Non determinisitic computation over the real numbers*, in "Philosophical Transactions of the Royal Society A", 2012, vol. 370, n^{o} 1971, p. 3349-3358, 10.1098/rsta.2011.0322.

[15] G. DOWEK. *The physical Church thesis as an explanation of the Galileo thesis*, in "Natural Computing", 2012, 10.1007/s11047-011-9301-x.

[16] G. DOWEK, M. J. GABBAY. *PNL to HOL: From the logic of nominal sets to the logic of higher-order functions*, in "Theoretical Computer Science", 2012, vol. 451, p. 38-69.

[17] G. DOWEK, O. HERMANT. *A Simple Proof That Super-Consistency Implies Cut Elimination*, in "Notre-Dame Journal of Formal Logic", 2012, vol. 53, n^{o} 4, p. 439-456, http://projecteuclid.org/euclid.ndjfl/1352383225.

[18] A. NARKAWICZ, C. MUÑOZ, G. DOWEK. *Provably Correct Conflict Prevention Bands Algorithms*, in "Science of Computer Programming", 2012, 10.1016/j.scico.2011.07.002.

International Conferences with Proceedings

[19] P. ARRIGHI, A. DÍAZ-CARO, B. VALIRON. *A type system for the vectorial aspects of the linear-algebraic lambda-calculus*, in "Proceedings of the 7th International Workshop on Developments of Computational Methods (DCM 2011)", E. KASHEFI, J. KRIVINE, F. VAN RAAMSDONK (editors), Electronic Proceedings in Theoretical Computer Science, Open Publishing Association, 2012, vol. 88, p. 1–15.

[20] P. AYRAULT, V. BENAYOUN, C. DUBOIS, F. PESSAUX. *ML Dependency Analysis for Assessors*, in "Software Engineering and Formal Methods - 10th International Conference, SEFM 2012 2012. Proceedings", Thessaloniki, Greece, G. ELEFTHERAKIS, M. HINCHEY, M. HOLCOMBE (editors), Lecture Notes in Computer Science, Springer, October 1-5 2012, vol. 7504, p. 278-292.

[21] M. BOESPFLUG, G. BUREL. *CoqInE: Translating the Calculus of Inductive Constructions into the λΠ-calculus Modulo*, in "Second International Workshop on Proof Exchange for Theorem Proving", D. PICHARDIE, T. WEBER (editors), 2012.

[22] M. BOESPFLUG, Q. CARBONNEAUX, O. HERMANT. *The λΠ-calculus Modulo as a Universal Proof Language*, in "Second Workshop on Proof Exchange for Theorem Proving (PxTP)", CEUR-WS.org, 2012, vol. 878, Available at: ceur-ws.org/Vol-878/paper2.pdf.

[23] P. BUIRAS, A. DÍAZ-CARO, M. JASKELIOFF. *Confluence via strong normalisation in an algebraic λ-calculus with rewriting*, in "Proceedings 6th Workshop on Logical and Semantic Frameworks with Applications (LSFA 2011)", S. RONCHI DELLA ROCCA, E. PIMENTEL (editors), Electronic Proceedings in Theoretical Computer Science, Open Publishing Association, 2012, vol. 81, p. 16–29.

[24] M. CARLIER, C. DUBOIS, A. GOTLIEB. *A Certified Constraint Solver over Finite Domains*, in "FM 2012: Formal Methods - 18th International Symposium", Paris, France, D. GIANNAKOPOULOU, D. MÉRY (editors), Lecture Notes in Computer Science, Springer, August 27-31 2012, vol. 7436, p. 116-131.

[25] M. CARLIER, C. DUBOIS, A. GOTLIEB. *A First Step in the Design of a Formally Verified Constraint-Based Testing Tool: FocalTest*, in "Tests and Proofs - 6th International Conference, TAP 2012", Prague, Czech

Republic, A. D. BRUCKER, J. JULLIAND (editors), Lecture Notes in Computer Science, Springer, May 31 - June 1 2012, vol. 7305, p. 35-50.

[26] D. COUSINEAU, O. HERMANT. *A Semantic Proof that Reducibility Candidates entail Cut Elimination*, in "RTA", A. TIWARI (editor), LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012, vol. 15, p. 133-148.

[27] G. DOWEK. *A theory independent Curry-De Bruijn-Howard correspondence*, in "International Colloquium on Automata, Languages and Programming", 2012, invited talk.

[28] G. DOWEK. *Around the physical Church-Turing thesis: cellular automata, formal languages, and the principles of quantum theory*, in "Language and Automata Theory and Applications", Lecture Notes in Computer Science, 2012, vol. 7183, p. 21-37.

[29] G. DOWEK, M. J. GABBAY. *Nominal Semantics for Predicate Logic: Algebras, Substitution, Quantifiers, and Limits*, in "Italian Convention of Computational Logic", 2012.

[30] A. DÍAZ-CARO, G. DOWEK. *Non determinism through type isomophism*, in "Proceedings of the 7th Workshop on Logical and Semantic Frameworks with Applications (LSFA 2012)", Rio de Janeiro, Brazil, 2012, To appear in EPTCS. Preprint at http://www.diaz-caro.info/ndti.pdf.

[31] A. DÍAZ-CARO, G. MANZONETTO, M. PAGANI. *Call-by-value non-determinism in a linear logic type discipline*, in "Logical Foundations of Computer Science, International Symposium, (LFCS 2013)", S. ARTEMOV, A. NERODE (editors), Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2013, vol. 7734, p. 164–178, To appear. Preprint at http://www.diaz-caro.info/dmp12.pdf.

[32] A. DÍAZ-CARO, B. PETIT. *Linearity in the non-deterministic call-by-value setting*, in "Logic, Language, Information and Computation", L. ONG, R. DE QUEIROZ (editors), Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2012, vol. 7456, p. 216–231.

[33] M. JACQUEL, K. BERKANI, D. DELAHAYE, C. DUBOIS. *Tableaux Modulo Theories Using Superdeduction - An Application to the Verification of B Proof Rules with the Zenon Automated Theorem Prover*, in "Automated Reasoning - 6th International Joint Conference, IJCAR 2012", Manchester, UK, Lecture Notes in Computer Science, Springer, June 26-29 2012, vol. 7364, p. 332-338.

[34] T. G. LE, O. HERMANT, M. MANCENY, R. PAWLAK, R. RIOBOO. *Unifying Event-based and Rule-based Styles to Develop Concurrent and Context-aware Reactive Applications - Toward a Convenient Support for Concurrent and Reactive Programming*, in "ICSOFT", S. HAMMOUDI, M. VAN SINDEREN, J. CORDEIRO (editors), SciTePress, 2012, p. 347-350.

[35] P. NÉRON. *A Formal Proof of Square Root and Division Elimination in Embedded Programs*, in "Certified Programs and Proofs - Second International Conference, CPP 2012", Kyoto, Japan, C. HAWBLITZEL, D. MILLER (editors), Lecture Notes in Computer Science, Springer, December 13-15 2012, vol. 7679, p. 256-272.

[36] P.-N. TOLLITTE, D. DELAHAYE, C. DUBOIS. *Producing Certified Functional Code from Inductive Specifications*, in "Certified Programs and Proofs - Second International Conference, CPP 2012", Kyoto, Japan, C. HAWBLITZEL, D. MILLER (editors), Lecture Notes in Computer Science, Springer, December 13-15 2012, vol. 7679, p. 76-91.

Conferences without Proceedings

[37] G. DOWEK. *Chaos and the principle of a bounded density of information*, in "Physics and computation", 2012.

Scientific Books (or Scientific Book chapters)

[38] P. ARRIGHI, G. DOWEK. *The principle of a finite density of information*, in "Irreducibility and Computational Equivalence: Wolfram Science 10 Years After the Publication of A New Kind of Science", H. ZENIL (editor), Springer-Verlag, 2012.

[39] G. DOWEK, E. AL.. *Informatique et sciences du numérique - Spécialité ISN en terminale S*, Eyrolles, 2012.

Other Publications

[40] A. ASSAF. *Traduction de HOL en Dedukti*, MPRI, 2012.

[41] R. BOST. *Nombres réels et transformation de programmes*, École polytechnique, 2012.

[42] G. BUREL. *From Axioms to Rewriting Rules*, 2012, manuscript.

[43] Q. CARBONNEAUX. *Compilation JIT des termes de preuve*, MPRI, 2012.

[44] R. CAUDERLIER. *Object-Oriented Features in lambda-Pi-calculus modulo, Compiling FoCaLize to Dedukti*, MPRI, 2012.

[45] N. DERSHOWITZ, G. DOWEK. *Universality in Two Dimensions*, 2012, manuscript.

[46] G. DOWEK. *On the definition of the classical connectives and quantifiers*, 2012, manuscript.

References in notes

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

[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, p. 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, p. 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, p. 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, p. 292-298.

[53] P. MARTIN-LÖF. *Intuitionistic Type Theory*, Bibliopolis, 1984.

[54] D. PRAWITZ. *Natural Deduction: A Proof-Theoretical Study*, Almquist and Wiksell, 1965.