

RESEARCH CENTRE  
Saclay - Île-de-France

IN PARTNERSHIP WITH:  
ENS Paris-Saclay

2021  
ACTIVITY REPORT

Project-Team  
DEDUCTEAM

**DEDUCTEAM**

IN COLLABORATION WITH: Laboratoire de Méthodes Formelles

**DOMAIN**

Algorithmics, Programming, Software  
and Architecture

**THEME**

Proofs and Verification

# Contents

<b>Project-Team DEDUCTEAM</b>	<b>1</b>
<b>1 Team members, visitors, external collaborators</b>	<b>2</b>
<b>2 Overall objectives</b>	<b>3</b>
2.1 Objectives	3
2.2 History	3
<b>3 Research program</b>	<b>4</b>
3.1 Logical Frameworks	4
3.2 Interoperability and proof encyclopediae	4
3.3 Interactive theorem proving	4
3.4 Proof automation	4
<b>4 Application domains</b>	<b>5</b>
4.1 Interoperability	5
<b>5 Highlights of the year</b>	<b>5</b>
5.1 Awards	5
5.2 New software	5
5.2.1 Lambdapi	5
5.2.2 Dedukti	6
5.2.3 personoj	6
5.2.4 Agda2Dedukti	6
5.2.5 Coqine	7
5.2.6 Krajono	7
5.2.7 Holide	7
5.2.8 Logipedia	7
5.2.9 nubo	8
5.2.10 ekstrakto	8
5.2.11 Zenon Modulo	8
5.2.12 CoLoR	8
5.2.13 SizeChangeTool	9
5.2.14 SKonverto	9
<b>6 New results</b>	<b>9</b>
6.1 Development of Lambdapi	9
6.2 Theory of $\lambda\Pi$ -calculus modulo rewriting and other logical formalisms	10
6.3 New theories in the $\lambda\Pi$ -calculus modulo rewriting	10
6.4 Translation from one theory to another	11
6.5 Automation for the Coq proof assistant	11
6.6 Theory of inductive definitions	12
6.7 Quantum Computing	12
<b>7 Bilateral contracts and grants with industry</b>	<b>13</b>
7.1 Bilateral contracts with industry	13
<b>8 Partnerships and cooperations</b>	<b>13</b>
8.1 International initiatives	13
8.1.1 STIC/MATH/CLIMAT AmSud project	13
8.2 International research visitors	14
8.2.1 Visits of international scientists	14
8.2.2 Visits to international teams	14
8.3 European initiatives	14
8.3.1 Horizon Europe	14

8.4 National initiatives	14
<b>9 Dissemination</b>	<b>14</b>
9.1 Promoting scientific activities	14
9.1.1 Scientific events: selection	14
9.1.2 Journal	15
9.1.3 Invited talks	15
9.1.4 Leadership within the scientific community	15
9.1.5 Scientific expertise	15
9.1.6 Research administration	15
9.2 Teaching - Supervision - Juries	15
9.2.1 Teaching	15
9.2.2 Supervision	16
9.3 Popularization	16
9.3.1 Internal or external Inria responsibilities	16
9.3.2 Articles and contents	17
<b>10 Scientific production</b>	<b>17</b>
10.1 Major publications	17
10.2 Publications of the year	17
10.3 Cited publications	19

## **Project-Team DEDUCTEAM**

*Creation of the Project-Team: 2017 January 01*

### **Keywords**

#### **Computer sciences and digital sciences**

A2.1.4. – Functional programming

A2.1.11. – Proof languages

A2.4.3. – Proofs

A3.1.1. – Modeling, representation

A7. – Theory of computation

A7.2. – Logic in Computer Science

#### **Other research topics and application domains**

B7. – Transport and logistics

## 1 Team members, visitors, external collaborators

### Research Scientists

- Gilles Dowek [Team leader, Inria, Senior Researcher, HDR]
- Pablo Arnault [Inria, Researcher, from Oct 2021]
- Bruno Barras [Inria, Researcher]
- Frédéric Blanqui [Inria, Researcher, HDR]
- Valentin Blot [Inria, Researcher]
- Jean-Pierre Jouannaud [Inria, Emeritus, HDR]
- Luidnel Maignan [Univ Paris-Est Marne La Vallée, Researcher, from Sep 2021]
- Renaud Vilmart [Inria, Starting Faculty Position]

### Post-Doctoral Fellow

- Pierre Vial [Inria]

### PhD Students

- Louise Dubois De Prisque [Inria]
- Mohamed Yacine El Haddad [CNRS, until Jul 2021]
- Thiago Felicissimo Cesar [Univ Paris-Saclay, from Oct 2021]
- Emilie Grienenberger [Ecole normale supérieure Paris-Saclay]
- Gabriel Hondet [Inria]
- Amelie Ledein [Inria]

### Technical Staff

- Boris Djalal [Inria, Engineer]

### Interns and Apprentices

- Pauline Bonnet [CNRS, from Sep 2021]
- Aurelien Castre [Inria, from Jul 2021 until Aug 2021]
- Corentin Chabanol [Centrale Supélec, Intern, L3]
- Luc Chabassier [Inria, from Apr 2021 until Jul 2021]
- Loris Cros [Centrale Supélec, Intern, M1]
- Thiago Felicissimo Cesar [Inria, from Mar 2021 until Aug 2021]
- Yoan Geran [Inria, from Mar 2021 until Aug 2021]
- Yann Leray [Inria, from Mar 2021 until Jul 2021]
- Thomas Traversié [Centrale Supélec, Intern, M1]
- Oleksii Tsokurov [Inria, from May 2021 until Aug 2021]

## Administrative Assistant

- Alexandra Merlin [Inria]

## External Collaborators

- Pablo Arrighi [Univ Paris-Saclay, HDR]
- Guillaume Burel [Ecole nationale supérieure d'informatique pour l'industrie et l'entreprise]
- Catherine Dubois [Ecole nationale supérieure d'informatique pour l'industrie et l'entreprise, HDR]
- Yoan Gérard [École Nationale Supérieure des Mines de Paris, From Sep 2021]
- Olivier Hermant [École Nationale Supérieure des Mines de Paris]
- Stéphane Lengrand [SRI International - USA, until Jul 2021]

## 2 Overall objectives

### 2.1 Objectives

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

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

### 2.2 History

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

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

The first version of **DEDUKTI** was developed in 2011 by Mathieu Boespflug [34]. From 2012 to 2015, new versions of **DEDUKTI** were developed and several theories were expressed in **DEDUKTI**, allowing to import proofs developed in **MATITA** (with the tool **KRAJONO**), **HOL LIGHT** (with the tool **HOLIDE**), **FOCALIZE** (with the tool **FOCALIDE**), **IPROVER**, and **ZENON**, totalizing several hundred of megabytes of proofs. The development of **DEDUKTI** has now been completed.

We now focus on the translation of proofs from one **DEDUKTI** theory to another and to the exporting of proofs to other proof systems. In particular the **MATITA** arithmetic library has been translated to a much weaker theory: constructive simple type theory, allowing to export it to **COQ**, **LEAN**, **PVS**, **HOL LIGHT**, and **ISABELLE/HOL**. In the same way, the first book of Euclids elements, formalized in **COQ** has

been exported to these systems as well. This led us to develop an online proof repository NUBO and an on-line encyclopedia LOGIPEDIA, allowing to share and browse this library.

We also focus on the development of new theories in DEDUKTI.

Finally, we develop an interactive theorem prover LAMBDAPI based on the same formalism as DEDUKTI. This interactive theorem prover is also used as a tool in the process of translating proofs from PVS and from automated theorem provers.

## 3 Research program

### 3.1 Logical Frameworks

A thesis, which is at the root of our research effort, is that logical systems should be expressed as theories in a logical framework. As a consequence, proof-checking systems should not be focused on one theory, such as Simple type theory, Martin-Löf's type theory, or the Calculus of constructions, but should be theory independent. On the more theoretical side, the proof search algorithms, or the algorithmic interpretation of proofs should not depend on the theory in which proofs are expressed, but this theory should just be a parameter. This is for instance expressed in the title of our invited talk at ICALP 2012: *A theory independent Curry-De Bruijn-Howard correspondence* [36].

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

The  $\lambda\Pi$ -calculus modulo theory, that is implemented in the system DEDUKTI and that is a synthesis of the Edinburgh logical framework and of Deduction modulo theory, subsumes them all.

### 3.2 Interoperability and proof encyclopaediae

Using a single prover to check proofs coming from different systems naturally leads to investigate how these proofs can be translated from one theory to another and used in a system different from the system in which they have been developed. This issue is of prime importance because developments in proof systems are getting bigger and, unlike other communities in computer science, the proof checking community has given little effort in the direction of standardization and interoperability.

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

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

### 3.3 Interactive theorem proving

We also want to investigate how the  $\lambda\Pi$ -calculus modulo theory can be used as the basis of an interactive theorem prover. This leads to two new scientific questions: first, how much can a tactic system be theory independent, and then how does rewriting extends the possibility to write tactics.

This has led to the development of LAMBDAPI, which is an interactive theorem prover for the  $\lambda\Pi$ -calculus modulo theory. Several tactics have been developed for this system, which are intended to help a human user to write proofs in our system instead of writing proof terms by hand.

### 3.4 Proof automation

Interoperability between interactive and automatic theorem provers can be fruitful to both systems: results coming from automatic solvers can be checked by a third-party software with an identified kernel,

and interactive provers can benefit from more automation. We are pushing towards this last application by extending the **SMTCoq plugin** for the Coq proof assistant with new logical transformations that encode Coq goals into first-order logic, which is the input logic of the class of automatic provers called SMT solvers.

## 4 Application domains

### 4.1 Interoperability

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

We view our work on interoperability and on the design of a formal proof encyclopedia as a service to the formal proof community.

## 5 Highlights of the year

### 5.1 Awards

Frédéric Blanqui's European COST project **EuroProofNet** has been accepted. EuroProofNet is the European research network on digital proofs. It aims at boosting the interoperability and usability of proof systems. It started on November 2021 and already gathers more than 220 researchers from 30 different countries. EuroProofNet organizes meetings and schools, and provides grants to its members for short-term scientific missions in another lab or country.

Frédéric Blanqui and Gabriel Hondet published a new release of the proof assistant **Lambdapi**. Lambdapi has been used in various works and, in particular, in Quentin Garchery's PhD [37].

Yacine El Haddad defended his PhD thesis on the integration of automated theorem provers in proof assistants [23].

Gilles Dowek and Alejandro Díaz-Caro have received the Best paper award (shared with another paper) at the 18th International Colloquium on Theoretical Aspects of Computing.

Renaud Vilmart received an accessit to the 2020 Gilles Kahn Ph.D. thesis award.

Yoan Gérard has received the Best student paper award at the 46th International Symposium on Mathematical Foundations of Computer Science.

### 5.2 New software

#### 5.2.1 Lambdapi

**Keywords:** Dependent types, Rewriting, Proof assistant

**Functional Description:** Lambdapi is an interactive proof development system featuring dependent types like in Martin-Löf's type theory, but allowing to define objects and types using oriented equations, aka rewriting rules, and reason modulo those equations. This allows to simplify some proofs, and formalize complex mathematical objects that are otherwise impossible or difficult to formalize in more traditional proof systems.

Lamdapi comes with Emacs and VSCode support.

Lamdapi can also read and output Dedukti files, and can thus be used as an higher-level intermediate language for translating proofs from one system to Dedukti.

Lamdapi is a logical framework and does not come with a pre-defined logic. However, it is easy to define a logic by declaring a few symbols and rules. A library of pre-defined logic is also provided.

Here are some of the features of Lambdapi: - Emacs and VSCode plugins (based on LSP) - support for unicode (UTF-8) and user-defined infix operators - symbols can be declared commutative, or associative and commutative - some arguments can be declared as implicit: the system will try to find out their value automatically - symbol and rule declarations are separated so that one can easily define inductive-recursive types or turn a proved equation into a rewriting rule - support for



interactive resolution of typing goals, and unification goals as well, using tactics - a rewrite tactic similar to the one of `SSReflect` in `Coq` - the possibility of calling external automated provers - a command is provided for automatically generating an induction principle for (mutually defined) strictly-positive inductive types - `Lambdapi` can call external provers for checking the confluence and termination of user-defined rewriting rules by translating them to the `XTC` and `HRS` formats used in the termination and confluence competitions

**URL:** <https://github.com/Deducteam/lambdapi>

**Contact:** Frederic Blanqui

### 5.2.2 Dedukti

**Keyword:** Logical Framework

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

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

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

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

**URL:** <https://deducteam.github.io/>

**Publications:** [hal-01086609](#), [hal-01176715](#), [hal-01441751](#)

**Contact:** François Thiré

**Participants:** François Thiré, Gaspard Ferey, Guillaume Genestier, Rodolphe Lepigre

### 5.2.3 personoj

**Keywords:** PVS, Automated theorem proving, Dedukti, Machine translation

**Functional Description:** `Personoj` comprises a set of PVS patches that may be used to export PVS specifications (propositions and definitions) or to export successive sequents of a proof to `lambdapi`. Another program is able to process these sequents and call automated theorem provers through `Why3` to prove the implications of the successive sequents.

**Contact:** Gabriel Hondet

### 5.2.4 Agda2Dedukti

**Keywords:** Compilation, Proof assistant, Higher-order logic, Rewriting systems

**Functional Description:** Translation of `Agda` proofs to the Logical Framework `Dedukti`.

**URL:** <https://github.com/Deducteam/Agda2Dedukti>

**Contact:** Guillaume Genestier

**Partner:** Chalmers University

### 5.2.5 Coqine

**Name:** Coq In dEdukti

**Keywords:** Higher-order logic, Formal methods, Proof

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

**URL:** [http://www.ensie.fr/~guillaume.burel/blackandwhite\\_coqInE.html.en](http://www.ensie.fr/~guillaume.burel/blackandwhite_coqInE.html.en)

**Contact:** Guillaume Burel

### 5.2.6 Krajono

**Keyword:** Proof

**Functional Description:** Krajono translates Matita proofs into Dedukti[CiC] (encoding of CiC in Dedukti) terms.

**Contact:** François Thiré

### 5.2.7 Holide

**Keyword:** Proof

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

**URL:** <https://github.com/Deducteam/Holide>

**Contact:** Guillaume Burel

### 5.2.8 Logipedia

**Name:** Logipedia

**Keywords:** Formal methods, Web Services, Logical Framework

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

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

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

**Release Contributions:** This is the beta version of Logipedia. It implements the functionalities mentioned above.

**URL:** <http://www.logipedia.science>

**Contact:** François Thiré

### 5.2.9 nuboo

**Name:** Nuboo

**Keywords:** Interoperability, Proof

**Functional Description:** Nuboo is a repository of formal proofs for computer scientists and mathematicians. Nuboo aims to leverage the interoperability issues raised by the substantial quantity of proof systems. To do so, it relies on a formalism in which many proofs of other systems can be stated. This formalism allows to translate formal developments to and fro foreign systems. Nuboo stores, classifies and serves those formal developments expressed in this general formalism. As such, developers may exchange their proofs, whatever their favourite system is.

**URL:** <https://github.com/Deducteam/nuboo>

**Contact:** Gabriel Hondet

### 5.2.10 ekstrakto

**Keywords:** TPTP, TSTP, Proof assistant, Dedukti

**Functional Description:** Extracting TPTP problems from a TSTP trace. Proof reconstruction in Dedukti from TSTP trace.

**URL:** <https://github.com/elhaddadyacine/ekstrakto>

**Contact:** Mohamed Yacine El Haddad

### 5.2.11 Zenon Modulo

**Keywords:** First-order logic, Automated theorem proving, Deduction Modulo

**Functional Description:** Zenon Modulo is an extension of the automated theorem prover Zenon. Compared to Super Zenon, it can 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.

**URL:** [https://github.com/Deducteam/zenon\\_modulo](https://github.com/Deducteam/zenon_modulo)

**Contact:** Pierre Halmagrand

### 5.2.12 CoLoR

**Name:** Coq Library on Rewriting and termination

**Keywords:** Coq, Formalisation

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

**URL:** <http://color.inria.fr/>

**Publications:** [inria-00543157](#), [inria-00390902](#), [inria-00084835](#)

**Authors:** Frederic Blanqui, Sebastien Hinderer

**Contact:** Frederic Blanqui

### 5.2.13 SizeChangeTool

**Keywords:** Rewriting systems, Proof assistant, Termination

**Functional Description:** A termination-checker for higher-order rewriting with dependent types.

Took part in the Termination Competition 2018 ([http://termination-portal.org/wiki/Termination\\_Competition\\_2018](http://termination-portal.org/wiki/Termination_Competition_2018)) in the "Higher-Order Rewriting (union Beta)" category.

**URL:** <https://github.com/Deducteam/SizeChangeTool>

**Contact:** Guillaume Genestier

**Partner:** Mines ParisTech

### 5.2.14 SKonverto

**Name:** SKonverto

**Keywords:** Skolemization, First-order logic, Proof assistant

**Functional Description:** SKonverto is a tool that transforms Lambdapi proofs containing Skolem symbols into proofs without these symbols.

**URL:** <https://github.com/Deducteam/SKonverto>

**Contact:** Mohamed Yacine El Haddad

## 6 New results

### 6.1 Development of Lambdapi

**Participants:** Frédéric Blanqui, Luc Chabassier, Loris Cros, Gabriel Hondet, Thomas Traversié.

Frédéric Blanqui and Gabriel Hondet published a new release of **Lamdapi**. Lambdapi is a proof assistant for the  $\lambda\Pi$ -calculus modulo rewriting. It extends Dedukti with meta-variables, implicit arguments, etc. and an interactive proof mode for solving typing and unification goals using tactics. Typing goals corresponding to first-order formulas can be discharged by calling external automated theorem provers via the Why3 library. Lambdapi can read and output Dedukti files and can thus be used as an intermediate higher-level language for translating proofs from one system to Dedukti.

Many improvements have been made on Lambdapi in 2021: parser generator changed to Menhir; generation of induction principles extended to parametrized strictly-positive types; addition of the tactics induction, admit and generalize; addition of associative and commutative symbols; addition of structured proof scripts (Aurélien Castre's internship), improvement of the Emacs and VSCode interface (Ashish Kumar Barnawal).

Lamdapi has been used in 2021 by:

- Luc Chabassier in his master internship to formalize a translation of **extensional type theory** in Martin-Löf's type theory extended with functional extensionality and uniqueness of identity proofs (Streicher's axiom K).
- Thomas Traversié in his internship to formalize **set theory**.
- Quentin Garchery in his **PhD** work for verifying Why3 transformation certificates [37]. He also developed a Lambdapi library on **binary integers**.

Loris Cros, in his internship supervised by Bruno Barras, has continued developing algorithms to check criteria related to confluence of rewrite systems in Dedukti. He also started the implementation of a completion algorithm in order to deal with rewriting modulo associativity and commutativity (AC).

## 6.2 Theory of $\lambda\Pi$ -calculus modulo rewriting and other logical formalisms

**Participants:** Frédéric Blanqui, Gilles Dowek, Jean-Pierre Jouannaud.

Gilles Dowek has investigated an alternative presentation of Pure Type Systems, where the notion of well-formed context has been dropped. This work has been published in the proceedings of Logical Frameworks and Meta-Languages [31].

Frédéric Blanqui has investigated and implemented in `Lambdapi` how to encode type universes without using matching modulo associativity and commutativity, which is complex to implement. It is based on a technique introduced in [33].

Gilles Dowek, Gaspard Férey, Jean-Pierre Jouannaud, Jiaxiang Liu have investigated confluence criterions for non-terminating Higher-Order Rewrite Theories.

## 6.3 New theories in the $\lambda\Pi$ -calculus modulo rewriting

**Participants:** Bruno Barras, Frédéric Blanqui, Valentin Blot, Luc Chabassier, Gilles Dowek, Thiago Felicissimo, Yoan Gérard, Émilie Grienenberger, Olivier Hermant, Gabriel Hondet, Amélie Ledein, Yann Leray, Thomas Traversié.

Frédéric Blanqui, Gilles Dowek, Émilie Grienenberger, Gabriel Hondet, and François Thiré have studied a theory in `Dedukti`, called `U`, that enables to express proofs in various theories (constructive and non constructive predicate logic, constructive and non constructive simple type theory, with and without polymorphism, with and without predicate subtyping, the Calculus of constructions, etc.) and shown that each of these theories corresponds to a selection of axioms in the theory `U`. This paper has been published in the proceedings of Formal Structures for Computation and Deduction [14].

During his Master internship, Luc Chabassier implemented in `lambdapi` a **translation from the Extensional Type Theory (ETT) to Intensional Type Theory (ITT)**, inspired from the paper "Eliminating Reflection From Type Theory" (Winterhalter, Sozeau et Tabareau) [40]. In contrast to what the authors did, his translation is well-typed by construction, rather than building an untyped term that has to be proved correct after the fact.

During his Master internship, Thiago Felicissimo worked on the translations from the proof assistant `Agda` to `Dedukti` and `Lambdapi`. In particular, he showed how to translate copattern matching coinduction [32], and implemented this in the tool `Agda2Dedukti`.

Gabriel Hondet and Frédéric Blanqui showed how to encode in the  $\lambda\Pi$ -calculus modulo rewriting, predicate subtyping and proof irrelevance, two important features of the `PVS` proof assistant. They proved that this encoding is correct and that encoded proofs can be mechanically checked by `Dedukti`, a type checker for the  $\lambda\Pi$ -calculus modulo theory using rewriting [38].

Luc Chabassier has started a Phd, supervised by Bruno Barras and Gilles Dowek, which goal is to study new approaches to formalize Category Theory in proof assistants, potentially exploiting the rewriting features of `Dedukti`.

Yoan Gérard has started a Phd, supervised by Olivier Hermant and Gilles Dowek, on reverse mathematics in `Dedukti`. His objective is to refine the translation framework from `Coq` to `Dedukti` with a focus on universes, to develop the corresponding tool, `Vodk`, along with reverse mathematics techniques to analyze and minimize the `Dedukti` proofs.

Thomas Traversié, during his internship supervised by Valentin Blot and Gilles Dowek, implemented set theory in `Lambdapi`. In particular, he encoded sets by a structure of pointed graphs, following the lines given in the paper "Cut elimination for Zermelo set theory" (Dowek and Miquel) [35].

Yann Leray improved and developed the translations from the proof assistant `Isabelle` to `Lambdapi` and `Dedukti` [30].

During her thesis, which started in October 2020, Amélie Ledein is interested in the verification of formal semantics proofs (such as the proof that a language is deterministic or the so-called subject

reduction property) of programming languages as well as the reuse of these proofs in another proof assistant. A first result obtained concerns the translation of K, a framework for defining formal semantics of programming languages from which different tools are automatically derived, into Dedukti. The implementation of this translation, named KaMeLo, is available [here](#). This work was presented during an LVP research day in 2021, and will be published at the JFLA national conference in June 2022.

## 6.4 Translation from one theory to another

**Participants:** Frédéric Blanqui, Guillaume Burel, Yacine El Haddad, Gabriel Hondet.

Gabriel Hondet investigated how to translate theories with implicit subtyping into Lambdapi without translating the subtyping derivations. The framework has been modified to be able to synthesise back the subtyping derivations using a generic coercion insertion algorithm. The mechanism has been implemented in the tool [Personoj](#) and tested on the translation of the standard library of PVS.

Yacine El Haddad, Guillaume Burel and Frédéric Blanqui showed the correctness of a de-skolemization transformation consisting in building a proof in a deep encoding of natural deduction in the  $\lambda\Pi$ -calculus modulo rewriting of a first-order formula from a proof of its skolemization. This transformation has been implemented in the tool [SKonverto](#).

Yacine El Haddad defended his PhD thesis on the integration of automated theorem provers in proof assistants [23]. His contributions are:

- Calling external provers. Since the provers do not use the same logic and system as the proof assistant, we elaborated a translation between Lambdapi's logic which is the  $\lambda\Pi$ -calculus Modulo Theory and the provers' logic which is first-order logic. This translation is implemented inside Lambdapi as part of its tactic language. Moreover, we showed that this translation is correct, i.e., if there is a proof of the translation of a formula in first-order logic, then there is a proof of that formula in the  $\lambda\Pi$ -calculus Modulo Theory.
- Reconstructing proofs. Proof assistants do not always support the output of external provers. However, for some proof assistants, there is a prover that outputs a format supported by that proof assistant. This is the case of Lambdapi with ZenonModulo and ArchSAT, and Coq with Zenon. We designed an architecture to reconstruct proofs generated by provers into Lambdapi by detecting which information is missing and asking a prover supported by Lambdapi to generate that missing information. The proposed solution does not depend on a specific prover and can reconstruct a considerable amount of proofs. The output format of the provers chosen is TSTP since it is the standard format used to test automated theorem provers. This solution is implemented as an independent tool named [Ekstrakto](#).
- Transforming Skolemized proofs. Provers can use many techniques to prove a formula, and one of these techniques is Skolemization. This transformation step transforms the original formula into an equi-satisfiable but not syntactically equivalent formula. We implemented an algorithm that transforms the proof of the new formula into a proof of the original one. The presented algorithm is proved correct and could be used separately if the proof fits a particular specification. This algorithm is implemented as a tool named [SKonverto](#), which could be used with Ekstrakto to build proofs in Lambdapi.

Yoan Gérard, during his internship, has analyzed the [GeoCoq](#) proof library, that contain the proofs of Euclid's Elements, book I. He has shown that it can be shrunk from the Calculus of Constructions with universes to fit in a logic as weak as predicate logic. Through the tools developed at Deducteam, this allowed GeoCoq to be exported to 7 systems : HOL Light, Lean, Matita, OpenTheory (hence Isabelle/HOL and HOL4), and PVS.

## 6.5 Automation for the Coq proof assistant

**Participants:** Valentin Blot, Louise Dubois de Prisque, Pierre Vial.

In order to automatize the Coq proof assistant, tactics which send a first-order goal to SMT solvers are available in the SMTCoq plugin.

Valentin Blot, Louise Dubois de Prisque and Pierre Vial, with the external collaboration of Chantal Keller, developed a new Coq automatic tactic [15] which generates and proves first-order statements about Coq terms (datatypes and functions). This enriches the semantics of information transmitted to SMTCoq and increases automation in the Coq proof assistant.

This tactic `snipe` is modular and combines independent transformations, which allows incremental development.

## 6.6 Theory of inductive definitions

**Participants:** Catherine Dubois, Amélie Ledein.

Catherine Dubois, Amélie Ledein and Mathieu Montin (Loria) have developed the `LIBNDT` library, implemented both in Agda and Coq, concerning a class of nested inductive types. This library proposes a generic type LNDT (linked nested datatype) of which some common data types (List, Maybe, Nest and even Bush - in Agda only for the latter) can be constructed as various instances. A paper has been accepted for publication in the international workshop MSFP 2022 (Mathematically Structured Functional Programming).

## 6.7 Quantum Computing

**Participants:** Pablo Arnault, Gilles Dowek, Renaud Vilmart.

Gilles Dowek and Alejandro Díaz-Caro have investigated an extension of propositional logic with a new connective, called `sup`, that has the introduction rules of the conjunction and the elimination rules of the disjunction and shown that this connective enables to model non-deterministic and non-reversible processes such as quantum measurement. This work [25] has been published in the proceedings of the 18th International Colloquium on Theoretical Aspects of Computing, where it has received the Best paper award (shared with another paper).

Renaud Vilmart, together with collaborators Benoît Valiron and Kostia Chardonnet have laid the foundations for a geometry of interaction-style of semantics for graphical language ZX-Calculus, devoted to quantum computing. This paper has been published in the proceedings of MFCS 2021 [17], and also in the proceedings of TLLA 20121 [16].

Renaud Vilmart has investigated the links between graphical language ZH-Calculus and Quantum Multiple-valued Decision Diagrams (QMDDs) used in verification of quantum processes. This paper has been published in the proceedings of MFCS 2021 [19].

Renaud Vilmart has published a paper at FoSSaCS 2021 [39]. This paper does not appear in the team's publications, as it was submitted before the author joined Deducteam.

Pablo Arnault has introduced a discrete-time quantum walk (DQW) defined on a polar grid, a primer, and are hence able to define an angular momentum for this DQW, which is conserved at the discrete level, i.e., on the spacetime lattice. He has shown that in the continuum limit the standard relativistic Landau levels of the Dirac equation are recovered in the case the Hamiltonian has been codiagonalized with, not the linear momentum, but the angular one. Those results have been published in [13].

Pablo Arnault has shown that one can discretize the Dirac equation under the form of a DQW without specifying a particular representation of the Clifford algebra. A new Clifford algebra has been defined from the operators defining the DQW, by requiring that the square of the DQW delivers what we define

as a valid discretization of the Dirac equation (this parallels exactly, at the discrete level, Dirac's original procedure in the continuum). We then show that the DQW contains, natively, a Wilson term that enables to avoid fermion doubling, and this without breaking unitarity, while discrete-time versions of standard lattice gauge theory are usually formulated in a Lagrangian way so that unitarity is not in-built, has to be proven, and is actually frequently broken. Those results appear in [24].

## 7 Bilateral contracts and grants with industry

### 7.1 Bilateral contracts with industry

**Participants:** Valentin Blot.

Valentin Blot and Chantal Keller have funding for a 4-year project (2021–2025) involving a PhD student, a research engineer (2 years) and a post-doctoral researcher (2 years). This funding is part of the Inria - Nomadic labs partnership for Tezos blockchain.

## 8 Partnerships and cooperations

### 8.1 International initiatives

#### 8.1.1 STIC/MATH/CLIMAT AmSud project

**Participants:** Gilles Dowek.

#### QAPLA

**Title:** Quantum aspects of programming languages

**Local supervisor:** Gilles Dowek

**Begin date:** Fri Jan 01 2021

**End date:** Sat Dec 31 2022

**Partners:**

- Universidad de Buenos Aires
- Universidad de la Republica Uruguay
- Universidad de Chile

**Summary:** The design of quantum programming languages is a rich framework that allows studying intrinsic properties of the computation we are modelling, such as parallelism, entanglement, superposition, etc; also, it is a way to study new logics (quantum logics with a computational ground), as well as to study classical logics from a new perspective. Finally, the studying the foundational bases of programming languages gives a path to develop proper implementations. This project proposes to study several aspects of quantum programming languages, with different approaches (quantum control/classical data, quantum control and data, categorical techniques, semantical techniques, realizability). The final aim is to merge different approaches in order to study from logics to implementations.



## 8.2 International research visitors

### 8.2.1 Visits of international scientists

Alejandro Díaz-Caro visited Deducteam for two weeks in November.

### 8.2.2 Visits to international teams

**Research stays abroad** Gilles Dowek has visited the University of Buenos Aires for two weeks.

Gilles Dowek has visited the Pontificia Universidade Católica do Rio de Janeiro for two weeks. He has given two talks there "A New connective in natural deduction, and its application to quantum computing" and "Ecumenism: logical constants and beyond".

## 8.3 European initiatives

### 8.3.1 Horizon Europe

**Participants:** Frédéric Blanqui.

**EuroProofNet** Frédéric Blanqui is the chair of the European COST action CA20111 **EuroProofNet**. EuroProofNet is the European research network on digital proofs. It aims at boosting the interoperability and usability of proof systems. It started on November 2021 and already gathers more than 220 researchers from 30 different countries. EuroProofNet organizes meetings and schools, and provides grants to its members for short-term scientific missions in another lab or country.

## 8.4 National initiatives

**Participants:** Gilles Dowek, Catherine Dubois.

The ANR project (2022-2025) ICSPA (Interoperable and Confident Set-based Proof Assistants) has been accepted in the context of the AAPG 2021 call. It is coordinated by Catherine Dubois and has the following academic partners Samovar – Inria Grand Est – Inria Paris-Saclay – LIRMM – IRIT with the industrial partner Clearys. The project starts on January 1st 2022. This project aims at reinforcing the confidence in proofs carried out mechanically for the set-based specification formalisms B, Event-B, and TLA+ that are used in industry. This will be done by verifying these proofs formally and independently with the proof verifier Dedukti. The project also aims at designing and implementing an exchange framework, through which those three systems can share their proofs and theories, making them effectively interoperable.

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

## 9 Dissemination

### 9.1 Promoting scientific activities

#### 9.1.1 Scientific events: selection

##### Member of the conference program committees

Frédéric Blanqui was PC member of **FSCD'21**, **CICM'21** and **TYPES'21**.

Gilles Dowek has been a PC member of CADE, ITP, Happoc, and NFR.

## Reviewer

Renaud Vilmart has reviewed papers for Logics in Computer Science (LICS), Quantum Physics and Logics (QPL) and Symposium on Theory of Computing (STOC).

### 9.1.2 Journal

#### Reviewer - reviewing activities

Renaud Vilmart has reviewed papers for Journal of the ACM (JACM), Logical Methods in Computer Science (LMCS), Physical Review A (PRA), Theoretical Computer Science (TCS) and Transactions on Quantum Computing (TQC).

### 9.1.3 Invited talks

Gilles Dowek has given an invited talk "Sharing proofs across logics and systems: a boost for formal methods?" at ABZ 2021.

Gilles Dowek has given an invited talk "Sharing geometry proofs across logics and systems" at the 13th International Conference on Automated Deduction in Geometry.

Gilles Dowek has given an invited talk "A framework to express the axioms of mathematics" at the VI congreso latinoamericano de matemáticos.

Gilles Dowek has given an invited talk "How can we make formal proof teachable?" at Theorem Proving Components for Educational Software 2021.

Gilles Dowek has given an invited talk "Explanation: from ethics to logic" at the Symposium Fairness, Integrity and Transparency in Formal Systems: Challenges for a Society Increasingly Dominated by Technology, co-organised by the Japan Association for Philosophy of Science and the Division for Logic, Methodology and Philosophy of Science and Technology of the International Union of History and Philosophy of Science and Technology. The video is online.

Bruno Barras has given an invited talk "Type Theory in Set Theory, in Type Theory" at the conference celebrating the 90 Years of Gödel's Incompleteness Theorems, held in Nürtingen (Germany), in July 2021.

### 9.1.4 Leadership within the scientific community

Frédéric Blanqui is member of the steering committees of the [International School on Rewriting \(ISR\)](#), the international [TYPES](#) conference, and of the [ACM/IEEE Symposium on Logic in Computer Science \(LICS\)](#).

### 9.1.5 Scientific expertise

Gilles Dowek is a member of the of the Comité national pilote d'éthique du numérique.

Gilles Dowek is a member of the Conseil national du numérique.

Gilles Dowek is a member of the Scientific board of the Société informatique de France.

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

### 9.1.6 Research administration

Frédéric Blanqui is assistant director of the doctoral school ED STIC on computer science of the University Paris-Saclay.

## 9.2 Teaching - Supervision - Juries

### 9.2.1 Teaching

- Master: Renaud Vilmart, Calcul quantique avancé et codes correcteurs, 10h, M2, QDCS, France
- Master: Frédéric Blanqui, formal languages, 21h, M1, ENSIIE, France
- Master: Frédéric Blanqui, rewriting theory, 14h, M1, ENS Paris-Saclay, France

- Master: Bruno Barras, Proof Assistants, 12h, M2, MPRI, France
- Master: Gabriel Hondet, rewriting theory TD, 14h, M1, ENS Paris-Saclay, France
- License: Amélie Ledein, oriented-object programming (Scala) - project, 15h, L3, ENS Paris-Saclay, France
- License: Amélie Ledein, functional programming (OCaml) - project, 15h, L3, ENS Paris-Saclay, France
- License: Thiago Felicissimo, functional programming - TD, 20h, L3, Faculté des Sciences d'Orsay, France
- License: Thiago Felicissimo, logic - TD, 20h, L3, Faculté des Sciences d'Orsay, France
- License: Gabriel Hondet, operating systems and architecture TD, 22.5h, L3, ENS Paris-Saclay, France
- License: Gabriel Hondet, programming TD, 12h, L3, ENS Paris-Saclay, France
- License: Gabriel Hondet, projet bases de données, 22.5h, L3, ENS Paris-Saclay, France
- License: Renaud Vilmart, Projet informatique, 50h, L2, Faculté des Sciences d'Orsay, France
- License: Louise Dubois de Prisque, Logic, 22h
- License: Louise Dubois de Prisque, Compiling, 22h
- License: Louise Dubois de Prisque, Logical and algorithmic tools, 18h
- License: Bruno Barras, Introductory course to Intuitionistic Proofs, 6h, L3, University of Paris
- Gilles Dowek has taught an introductory course on Logic at the École normale de Paris-Saclay. For this flipped-classroom course, he has shot twelve one hour videos that are on-line "Douze leçons de logique".
- Gilles Dowek has organized a research workshop for students of the École normale de Paris-Saclay.
- IUT d'Orsay, 2nd year: Pierre Vial, Compiling, 44h
- Polytech Paris-Saclay, 4th year: Pierre Vial, hands-on in Functional programming (OCaml), 16h

### 9.2.2 Supervision

Frédéric Blanqui and Gilles Dowek advise the doctoral work of Gabriel Hondet.

Frédéric Blanqui and Gilles Dowek advise the doctoral work of Thiago Felicissimo.

Bruno Barras and Gilles Dowek advise the doctoral work of Luc Chabassier.

Gilles Dowek and Olivier Hermant advise the doctoral work of Yoan Gérard.

Frédéric Blanqui and Gilles Dowek have advised the masters internship of Thiago Felicissimo.

Gilles Dowek and Olivier Hermant have advised the masters internship of Yoan Gérard.

Gilles Dowek has advised the masters internship of Oleksii Tsokurov.

Valentin Blot and Gilles Dowek advise the parcours recherche of Thomas Traversié.

Gilles Dowek and Jean-Pierre Jouannaud advise the parcours recherche of Corentin Chabanol.

Bruno Barras advise the parcours recherche of Loris Cros.

## 9.3 Popularization

### 9.3.1 Internal or external Inria responsibilities

Frédéric Blanqui is member of the Evaluation Committee of Inria.

Frédéric Blanqui is member of the scientific committee of Inria Saclay.

### 9.3.2 Articles and contents

Renaud Vilmart has written an article for the blog “Binaire du Monde”, related to his PhD work.

## 10 Scientific production

### 10.1 Major publications

- [1] A. Assaf, G. Burel, R. Cauderlier, D. Delahaye, G. Dowek, C. Dubois, F. Gilbert, P. Halmagrand, O. Hermant and R. Saillard. ‘Expressing theories in the  $\lambda\Pi$ -calculus modulo theory and in the Dedukti system’. In: *22nd International Conference on Types for Proofs and Programs, TYPES 2016*. Novi SAD, Serbia, May 2016. URL: <https://hal-mines-paristech.archives-ouvertes.fr/hal-01441751>.
- [2] B. Barras, T. Coquand and S. Huber. ‘A generalization of the Takeuti-Gandy interpretation’. In: *Mathematical Structures in Computer Science* 25.5 (2015), pp. 1071–1099. DOI: [10.1017/S0960129514000504](https://doi.org/10.1017/S0960129514000504). URL: <https://doi.org/10.1017/S0960129514000504>.
- [3] F. Blanqui. ‘Definitions by rewriting in the Calculus of Constructions’. Anglais. In: *Mathematical Structures in Computer Science* 15.1 (2005), pp. 37–92. DOI: [10.1017/S0960129504004426](https://doi.org/10.1017/S0960129504004426). URL: <http://hal.inria.fr/inria-00105648/en/>.
- [4] F. Blanqui, J.-P. Jouannaud and A. Rubio. ‘The Computability Path Ordering’. In: *Logical Methods in Computer Science* (Oct. 2015). DOI: [10.2168/LMCS-11\(4:3\)2015](https://doi.org/10.2168/LMCS-11(4:3)2015). URL: <https://hal.inria.fr/hal-01163091>.
- [5] V. Blot. ‘An interpretation of system F through bar recursion’. In: *32nd ACM/IEEE Symposium on Logic in Computer Science*. IEEE, 2017.
- [6] G. Burel, G. Bury, R. Cauderlier, D. Delahaye, P. Halmagrand and O. Hermant. ‘First-Order Automated Reasoning with Theories: When Deduction Modulo Theory Meets Practice’. In: *Journal of Automated Reasoning* (2019). DOI: [10.1007/s10817-019-09533-z](https://doi.org/10.1007/s10817-019-09533-z). URL: <https://hal.archives-ouvertes.fr/hal-02305831>.
- [7] D. Cousineau and G. Dowek. ‘Embedding Pure Type Systems in the  $\lambda\Pi$ -calculus modulo’. In: *Typed lambda calculi and applications*. Ed. by S. R. della Rocca. Vol. 4583. Lecture Notes in Computer Science. Springer-Verlag, 2007, pp. 102–117.
- [8] G. Dowek, T. Hardin and C. Kirchner. ‘Theorem proving modulo’. In: *Journal of Automated Reasoning* 31 (2003), pp. 33–73.
- [9] O. Hermant. ‘Resolution is Cut-Free’. In: *Journal of Automated Reasoning* 44.3 (Mar. 2010), pp. 245–276.
- [10] M. Jacquél, K. Berkani, D. Delahaye and C. Dubois. ‘Tableaux Modulo Theories Using Superdeduction’. In: *Global Journal of Advanced Software Engineering (GJASE)* 1 (Dec. 2014), pp. 1–13. DOI: [10.1007/978-3-642-31365-3\\_26](https://doi.org/10.1007/978-3-642-31365-3_26). URL: <https://hal.archives-ouvertes.fr/hal-01099338>.
- [11] M. Jacquél, K. Berkani, D. Delahaye and C. Dubois. ‘Verifying B Proof Rules using Deep Embedding and Automated Theorem Proving’. In: *Software and Systems Modeling (SoSyM)* (June 2013).

### 10.2 Publications of the year

#### International journals

- [12] G. Dowek, G. Férey, J.-P. Jouannaud and J. Liu. ‘Confluence of left-linear higher-order rewrite theories by checking their nested critical pairs’. In: *Mathematical Structures in Computer Science* (22nd Jan. 2022), pp. 1–36. DOI: [10.1017/S0960129522000044](https://doi.org/10.1017/S0960129522000044). URL: <https://hal.inria.fr/hal-03126111>.
- [13] G. Jay, P. Arnault and F. Debbasch. ‘Dirac quantum walks with conserved angular momentum’. In: *Quantum Studies: Mathematics and Foundations* 8.4 (Nov. 2021), pp. 419–430. DOI: [10.1007/s40509-021-00253-x](https://doi.org/10.1007/s40509-021-00253-x). URL: <https://hal.archives-ouvertes.fr/hal-03577256>.

### International peer-reviewed conferences

- [14] F. Blanqui, G. Dowek, É. Grienerberger, G. Hondet and F. Thiré. ‘Some Axioms for Mathematics’. In: FSCD 2021 - 6th International Conference on Formal Structures for Computation and Deduction. Buenos Aires / Virtual, Argentina, 17th July 2021. DOI: [10.4230/LIPIcs.FSCD.2021.20](https://doi.org/10.4230/LIPIcs.FSCD.2021.20). URL: <https://hal.inria.fr/hal-03279749>.
- [15] V. Blot, L. Dubois de Prisque, C. Keller and P. Vial. ‘General Automation in Coq through Modular Transformations’. In: Seventh Workshop on Proof Exchange in Theorem Proving. Proceedings of the Seventh Workshop on Proof eXchange for Theorem Proving. Pittsburgh, United States: EPTCS, 7th July 2021. DOI: [10.4204/EPTCS.336.3](https://doi.org/10.4204/EPTCS.336.3). URL: <https://hal.archives-ouvertes.fr/hal-03328935>.
- [16] K. Chardonnet, B. Valiron and R. Vilmart. ‘Geometry of Interaction for ZX-Diagrams’. In: TLLA 2021 - 5th International Workshop on Trends in Linear Logic and Applications. Rome / Virtual, Italy, 25th June 2021. URL: <https://hal-lirmm.ccsd.cnrs.fr/lirmm-03272411>.
- [17] K. Chardonnet, B. Valiron and R. Vilmart. ‘Geometry of Interaction for ZX-Diagrams’. In: MFCS 2021 - 46th International Symposium on Mathematical Foundations of Computer Science. Vol. 202. Leibniz International Proceedings in Informatics (LIPIcs). Tallinn, Estonia, 18th Aug. 2021, 30:1–30:16. DOI: [10.4230/LIPIcs.MFCS.2021.30](https://doi.org/10.4230/LIPIcs.MFCS.2021.30). URL: <https://hal.archives-ouvertes.fr/hal-03154573>.
- [18] M. Färber. ‘Safe, Fast, Concurrent Proof Checking for the lambda-Pi Calculus Modulo Rewriting’. In: 11th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP ’22). Philadelphia, PA, United States, 17th Jan. 2022. DOI: [10.1145/3497775.3503683](https://doi.org/10.1145/3497775.3503683). URL: <https://hal.inria.fr/hal-03143359>.
- [19] R. Vilmart. ‘Quantum Multiple-Valued Decision Diagrams in Graphical Calculi’. In: MFCS 2021 - 46th International Symposium on Mathematical Foundations of Computer Science. Vol. 202. Leibniz International Proceedings in Informatics (LIPIcs). Tallinn, Estonia, 23rd Aug. 2021, 89:1–89:15. DOI: [10.4230/LIPIcs.MFCS.2021.89](https://doi.org/10.4230/LIPIcs.MFCS.2021.89). URL: <https://hal.archives-ouvertes.fr/hal-03277262>.

### Conferences without proceedings

- [20] V. Blot, L. Dubois de Prisque, C. Keller and P. Vial. ‘Bécassine à la chasse au Coq’. In: JFLA 2022 - Journées Francophones des Langages Applicatifs. Saint-Médard-d’Excideuil, France, 28th June 2022. URL: <https://hal.archives-ouvertes.fr/hal-03604902>.
- [21] G. Férey and J.-P. Jouannaud. ‘Confluence in Non-Left-Linear Untyped Higher-Order Rewrite Theories’. In: PPDP 2021 - 23rd International Symposium on Principles and Practice of Declarative Programming. Tallin, Estonia, 6th Sept. 2021. DOI: [10.1145/3479394.3479403](https://doi.org/10.1145/3479394.3479403). URL: <https://hal.inria.fr/hal-03126115>.
- [22] A. Ledein, V. Blot and C. Dubois. ‘Vers une traduction de K en Dedukti’. In: JFLA 2022 - Journées Francophones des Langages Applicatifs. Saint-Médard-d’Excideuil, France, 28th June 2022. URL: <https://hal.archives-ouvertes.fr/hal-03604962>.

### Doctoral dissertations and habilitation theses

- [23] Y. El haddad. ‘Integrating Automated Theorem Provers in Proof Assistants’. Université Paris-Saclay, 9th Sept. 2021. URL: <https://tel.archives-ouvertes.fr/tel-03387912>.

### Reports & preprints

- [24] P. Arnault. *Clifford algebra from quantum automata and unitary Wilson fermions*. 16th Feb. 2022. URL: <https://hal.archives-ouvertes.fr/hal-03577279>.
- [25] A. Díaz-Caro and G. Dowek. *A New Connective in Natural Deduction, and its Application to Quantum Computing* ★. 23rd Jan. 2022. URL: <https://hal.inria.fr/hal-03540150>.

- [26] G. Férey and J.-P. Jouannaud. *Confluence in UnTyped Higher-Order Theories by means of Critical Pairs*. 14th Sept. 2021. URL: <https://hal.inria.fr/hal-03126102>.
- [27] J.-P. Jouannaud and F. Orejas. *Unification of Drags and Confluence of Drag Rewriting*. 31st Oct. 2021. URL: <https://hal.inria.fr/hal-02562152>.
- [28] J.-P. Jouannaud and F. Orejas. *Unification of Drags and Confluence of Drag Rewriting*. 1st Feb. 2021. URL: <https://hal.inria.fr/hal-02562463>.
- [29] A. Kissinger, R. Vilmart and J. van de Wetering. *Classical simulation of quantum circuits with partial and graphical stabiliser decompositions*. 11th Mar. 2022. URL: <https://hal.archives-ouvertes.fr/hal-03606226>.
- [30] Y. Leray. *M1 Internship Report Translating proofs between Isabelle and Dedukti*. Deducteam - Inria Paris-Saclay / ENS Paris-Saclay, 27th Sept. 2021. URL: <https://hal.inria.fr/hal-03348486>.

### Other scientific publications

- [31] G. Dowek. *Interacting Safely with an Unsafe Environment*. Pittsburgh, United States, 16th July 2021. DOI: [10.4204/eptcs.337.3](https://doi.org/10.4204/eptcs.337.3). URL: <https://hal.inria.fr/hal-03540149>.
- [32] T. Felicissimo. ‘Representing Agda and coinduction in the  $\lambda\Pi$ -calculus modulo rewriting’. LSV, ENS Paris Saclay, Université Paris-Saclay, 7th Sept. 2021. URL: <https://hal.inria.fr/hal-03343699>.

### 10.3 Cited publications

- [33] F. Blanqui, T. Hardin and P. Weis. ‘**On the implementation of construction functions for non-free concrete data types**’. In: *Proceedings of the 16th European Symposium on Programming*, Lecture Notes in Computer Science 4421. 15 pages. 2007.
- [34] M. Boespflug. ‘Conception d’un noyau de vérification de preuves pour le  $\lambda\Pi$ -calcul modulo’. PhD thesis. École Polytechnique, 2011.
- [35] G. Dowek and A. Miquel. ‘Cut elimination for Zermelo set theory’. In: *Manuscript, available from the web pages of the authors* (2007).
- [36] G. Dowek. ‘A Theory Independent Curry-de Bruijn-howard Correspondence’. In: *Proceedings of the 39th International Colloquium Conference on Automata, Languages, and Programming - Volume Part II*. ICALP’12. Warwick, UK: Springer-Verlag, 2012, pp. 13–15. DOI: [10.1007/978-3-642-31585-5-5\\_2](https://doi.org/10.1007/978-3-642-31585-5-5_2). URL: [http://dx.doi.org/10.1007/978-3-642-31585-5\\_5C\\_2](http://dx.doi.org/10.1007/978-3-642-31585-5_5C_2).
- [37] Q. Garchery. ‘**A Framework for Proof-carrying Logical Transformations**’. In: *Proceedings of the 7th International Workshop on Proof eXchange for Theorem Proving*, Electronic Proceedings in Theoretical Computer Science 336. 2021.
- [38] G. Hondet and F. Blanqui. ‘**Encoding of Predicate Subtyping and Proof Irrelevance in the  $\lambda\Pi$ -calculus Modulo Theory**’. In: *Proceedings of the 26th International Conference on Types for Proofs and Programs*, Leibniz International Proceedings in Informatics 188. 2021.
- [39] R. Vilmart. ‘The Structure of Sum-Over-Paths, its Consequences, and Completeness for Clifford’. In: *Foundations of Software Science and Computation Structures*. Ed. by S. Kiefer and C. Tasson. Cham: Springer International Publishing, 2021, pp. 531–550.
- [40] T. Winterhalter, M. Sozeau and N. Tabareau. ‘Eliminating reflection from type theory’. In: *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2019, Cascais, Portugal, January 14-15, 2019*. Ed. by A. Mahboubi and M. O. Myreen. ACM, 2019, pp. 91–103. DOI: [10.1145/3293880.3294095](https://doi.org/10.1145/3293880.3294095). URL: <https://doi.org/10.1145/3293880.3294095>.