Evaluation of Inria theme Proofs and Verification

Project-team πr^2

March 2019

Project-team title: Design, study and implementation of languages for proofs and programs

Scientific leader: Pierre-Louis Curien

Research centers: Paris

Common project-team with: CNRS, Université Paris 7

1 Personnel

Current composition of the project-team:

Research scientists and faculty members:
- Thierry Coquand, Prof. University of Gothenburg (INRIA chair)
- Pierre-Louis Curien, DR CNRS (team leader)
- Yves Guiraud, CR INRIA
- Hugo Herbelin, DR INRIA
- Pierre Letouzey, MCF Université Paris 7
- Jean-Jacques Lévy, DR INRIA (emeritus)
- Yann Régis-Gianas, MCF Université Paris 7
- Alexis Saurin, CR CNRS
- Matthieu Sozeau, CR INRIA

Engineers:
- Daniel de Rauglaudre, IR INRIA
- Thierry Martinez, IR INRIA (part-time)

Postdocs:
- Eric Finster, INRIA (until Feb. 2019)
- Kailiang Ji, INRIA (until Feb. 2019)
- Exequiel Rivas, INRIA (until Feb. 2019)

Ph.D. students:
- Antoine Allioux, INRIA (from Feb. 2018)
- Ahbishek De, FSMP (from Sep. 2018)
- Cédric Ho Thanh, Université Paris 7 (from Sep. 2017)
- Théo Zimmermann, Université Paris 7 (from Sep. 2016)

Administrative assistant:
- Mathieu Mourey, INRIA (from Sep. 2018)
- Sandrine Verges, INRIA (until June 2018)

Personnel at the start of the evaluation period (March 2015)

	INRIA	CNRS	University	Other	**Total**
DR (1) / Professors	1	1			**2**
DR emeritus	1				**1**
CR (2) / Assistant professors	2	1	2		**5**
Permanent engineers (3)					
Postdocs	1				**1**
PhD Students			10		**10**
Total	**5**	**2**	**12**		**19**

(1) "Senior Research Scientist (Directeur de Recherche)"
(2) "Junior Research Scientist (Chargé de Recherche)"
(3) "Civil servant (CNRS, INRIA, ...)"

Personnel at the time of the evaluation (March 2019)

	INRIA	CNRS	University	Other	**Total**
DR / Professors	1	1			**2**
DR emeritus	1				**1**
CR / Assistant professors	2	1	2		**5**
Permanent engineers	1				**1**
Post-docs					
PhD Students	1		2	1	**4**
Total	**6**	**2**	**4**	**1**	**13**

Changes in the scientific staff

The team composition in permanent research scientists and faculty members has been stable over the period. Daniel de Rauglaudre has joined as full time INRIA research engineer in June 2015.

Current position of former project-team members

- Cyrille Chenavier (PhD 2013-2016), currently postdoc at INRIA Lille
- Guillaume Claret (PhD 2012-2018), currently engineer at BlaBlaCar
- Amina Doumane (PhD 2014-2017), currently postdoc at Warsaw University
- Thibaut Girka (PhD 2015-2018), currently looking for a position
- Lourdes del Carmen González Huesca (PhD 2011-2015), currently teaching assistant at University of Mexico
- Guilhem Jaber (Postdoc ANR RAPIDO, 2016-2017), currently assistant professor at the University of Nantes
- Matej Košic (engineer funded by ADT Coq, 2015-2016), then with the Marelle team in 2016-2017
- Marc Lasson (Postdoc INRIA 2014-2015), currently engineer at Lexifi
- Gabriel Lewertowski (PhD 2015-2016), currently engineer at TrustInSoft
- Maxime Lucas (PhD 2014-2017), currently postdoc at INRIA Nantes
- Cyprien Mangin (PhD 2015-2018), currently engineer at Google Paris
- Étienne Miquey (PhD 2014-2017), currently postdoc at INRIA Nantes
- Jovana Obradović (PhD 2014-2017), currently postdoc at the Institute of Mathematics of the Czech Academy of Sciences, Prague
- Ludovic Patey (PhD 2012-2016), currently CR CNRS at Université Lyon 1

- Pierre-Marie Pédrot (PhD 2012-2015), currently CR INRIA at INRIA Nantes

Last INRIA enlistments

- Daniel de Rauglaudre, IR2, 2015 (internal transfer from INRIA Rocquencourt)
- Jean-Jacques Lévy, DR0 emeritus, 2014 (transfer from INRIA Saclay)
- Yves Guiraud, CR1, 2012 (transfer from INRIA Nancy), now CRCN
- Matthieu Sozeau, CR2, 2010 (hiring), now CRCN

Other comments:

The team will reach its planned end-date at the end of 2019, when the current scientific leader will retire. The team members are currently working on the collective scientific plans after that date.

Amina Doumane's PhD thesis (defended in June 2017) received the EACSL Ackermann award as well as Prix Gilles Kahn, attributed by the SIF. She also received the Kleene award of the best student paper at LICS 2017, as well as the "Prix scientifique de *La Recherche*" 2017 for the best French scientific paper in computer science.

Étienne Miquey received the Kleene award of the best student paper at LICS 2018. He also received the prize of the best thesis in mathematcis of Uruguay in 2018.

Matthieu Sozeau, together with Nicolas Tabareau (Inria Nantes) and Eric Tanter (Univ. Chile in Santiago) received the Distinguished Paper Award at ICFP 2018.

2 Research goals and results

2.1 Keywords

Proof theory, proof assistants, Coq, types, type theory, dependent types, intuitionistic, classical, and linear logic, control operators, delimited control, programming languages, dependently typed programming languages, effects, fixed-point logics, induction and coinduction, rewriting, polygraphs, higher categories, higher algebra, operas, homotopy.

2.2 Context and overall goals of the project

The πr^2 team studies the foundational aspects of formal proofs and programs, and develops the Coq proof assistant software – hosting the current coordinator of the development team –, with a focus on the dependently-typed programming language aspects of Coq. Since 2012, the team has also extended its scope to the effective aspects of higher algebra, which shares foundational tools with recent advances in type theory.

2.2.1 Proof theory and the Curry-Howard correspondence

Proof theory is the branch of logic devoted to the study of the structure of proofs. A significant early contributor to this field is Gentzen, who developed in 1935 two logical formalisms that are now central to the study of proofs [Gen35]. These are natural deduction, that is particularly well-suited to simulate the intuitive notion of reasoning, and sequent calculus, a syntax with deep geometrical properties that is particularly well-suited for proof automation. Proof theory gained a remarkable importance in computer science when it became clear, after genuine observations first by Curry [Cur34, CFC58], then later by Howard and de Bruijn at the end of the 60's [How80, dB68], that proofs had the very same structure as programs: for instance, natural deduction proofs can be identified as programs of the typed λ-calculus. An asymmetrical assignment of programs to proofs, named realisability, had been developed earlier by Kleene [Kle45], Gödel [G5̈8] and

Kreisel [Kre59] around the ideas of Brouwer. Altogether, realisability and the Curry-Howard correspondence between proofs and typed programs have been the starting point to a large spectrum of researches and results contributing to deeply connect logic and computer science.

The idea that proofs have computational contents has for long been limited to the intuitionistic case, until it was discovered in 1990 by Griffin [Gri90] that this connection was extensible to classical logic, proividing a stimulus for exploring new correspondences between computer science and logic. One of these, explored by Herbelin and Curien [Her95, CH00], was the computational understanding of Gentzen's sequent calculus, which not only makes a bridge with abstract machines in programming but also highlights symmetries between a program and its evaluation context, as well as between the standard notions of call-by-name and call-by-value evaluation semantics. Other explorations include: the question of the computational content of the classical axiom of choice, connecting it in particular to lazy evaluation and coinduction [Her12]; that of the computational content of the forcing method, connecting it to memory assignment; that of organising the computational content of connectives around concepts such as polarity and focusing.

2.2.2 Type Theory, development of Coq, and further research around proofs and programs

In 1971, Martin-Löf introduced a formalism, referred to as modern type theory, that is both a logical system and a typed functional programming language. The Calculus of Constructions was then designed in 1985 by Coquand and Huet [Coq85, CH85], based on Girard-Reynolds System F. This system served as the foundation of the first implementation of Coq in 1984. In 1989, Coquand and Paulin [CPM90] designed an extension of the Calculus of Constructions with a generalisation of algebraic datatypes called inductive types, leading to the Calculus of Inductive Constructions (CIC) that started to serve as a new foundation for Coq, which had its first public release, and is still at the core of the current system [99].

Over the course of 30 years of development, Coq has turned into a reference proof assistant, for the formalisation of mathematics and computer systems. The main ingredients of Coq are:

- A relatively small kernel, following the de Bruijn principle. Today, the kernel is essentially a type checker for the Polymorphic Calculus of Cumulative Inductive Constructions: it includes a standard dependent type theory, with universe polymorphism and cumulativity, mutual inductive and coinductive families, (co-)recursive definitions and case analysis. Additionally, the kernel includes efficient reduction machines for testing definitional equality of terms and a module system.
- A high-level language (Gallina) that can be elaborated down to the kernel, by means of a type inference algorithm relying on higher-order unification and a pattern-matching compiler, coercions, overloading mechanisms and notations.
- An interactive proof engine allowing to prove theorems by gradually refining a set of goals using various tactics and decision procedures.
- A set of libraries developing standard structures such as numbers, lists, finite sets and maps along with standard theories of sets, relations and orders.
- A toplevel system based on a state transition machine allowing to process proofs asynchronously, which is used by the CoqIDE GUI.
- An extraction procedure that maps programs (or even computational proofs) of the CIC to functional programs.

Our research in this broad field targets the following main aspects:

- The study and implementation of extended type theories, refining and providing more expressivity to the core calculus: this covers work on proof-irrelevance, the treatment of universes and the integration of concepts from homotopy type theory and proof theory such as the treatment of equality or effects in the system.

- Dependently-typed programming and proving: manipulation of dependently-typed structures in programs and proofs poses some theoretical and practical challenges. Our goal here is to provide high-level tools for the development of dependently-typed programs, their verification and efficient compilation.
- Since the last evaluation in 2015, we also developed two research directions: one is the study of infinitary proof systems, and the other is laying foundations for incremental programming, for which more details are provided in Sections 2.4 and 2.6, respectively.

2.2.3 Effective higher algebra

Higher structures originate in algebraic topology, where ∞-groupoids have been introduced to subsume the homotopy groups of a topological space X: the *fundamental ∞-groupoid* of X has the elements of X as 0-dimensional cells, the continuous paths of X as 1-cells, the homotopies between continuous paths as 2-cells, and so on. In the last decades, higher structures have appeared in other fields of mathematics, with the study of higher-dimensional versions of classical algebraic objects (e.g. in mathematical physics), or of weakened versions where equations hold only up to coherent equivalences (e.g. in representation theory). Recently, higher algebra has reached logic, leading to homotopy type theory (HoTT), when it was observed that the identity types are naturally equiped with a structure of ∞-groupoid: the 1-cells are the proofs of equality between two terms, the 2-cells are the proofs of equality between proofs of equality, and so on [Pro13]. However, higher structures are extremely complex objects, requiring new tools to reason about and to compute with them.

In order to describe higher categories combinatorially, *polygraphs* have been introduced as higher generalisations of presentations by generators and relations [Str76, Bur93]. Polygraphs led to a theory of higher rewriting, unifying usual rewriting-like paradigms, like abstract, word and term rewriting [Laf03, Mal04, Gui04, Gui06a]. Specific methods have been introduced to analyse the computational properties of polygraphs, based on algebraic concepts such as derivations to prove termination, which in turn gave new tools for complexity analysis [BG09]. Also, polygraphs allowed the generalisation of works by Anick [Ani86] and Squier [Squ87, SOK94], who computed homological and homotopical invariants of algebras and monoids by rewriting techniques. Since then, a deeper unified theory relating rewriting and homotopical algebra was established [GM09, GM11, GM12b, GM12a], yielding new rewriting methods in algebra. For example, rewriting algorithms can compute a coherent presentation of a monoid M, which is a combinatorial object that contains a presentation of M by generators and relations, together with higher cells that classify the different proofs of the same equalities: this is, in essence, the same as the proofs of equality of proofs of equality in homotopy type theory. When this process of "unfolding" proofs of equalities is pursued in every dimension, one gets a *polygraphic resolution* of M [Mé03, LMW10, AM11, GM12b], which is a candidate to be a faithful formalisation of M in homotopy type theory.

Our research in this field targets the following aspects:
- The development of formal representations of higher structures, including syntaxes to describe them and rewriting-like tools to compute normal forms.
- The application of rewriting methods to solve various problems about higher structures, such as the computation of coherence conditions or of resolutions, especially polygraphic resolutions.

2.3 Research axis 1: Proof theory, program semantics, side effects

2.3.1 Personnel

Permanent members: Pierre-Louis Curien, Hugo Herbelin, Yann Régis-Gianas, Matthieu Sozeau. Postdoc: Exequiel Rivas Gadda. PhD students: Etienne Miquey, Ludovic Patey, Pierre-Marie

Pédrot. Interns: Charlotte Barot, Kostia Chardonnet. External collaborator: Arnaud Spiwack (Tweag I/O).

2.3.2 Project-team positioning

This research axis is about the study of side effects in different contexts: of programming languages using general abstract notions of side effects such as monads, arrows or applicative functors; of program certification, to validate properties of effectful programs; of the proof-as-program correspondence, connecting side effects to logical translations, interpreting standard axioms as programs with side effects. Sometimes, to compare the logical or computational strength of logical statements, reverse mathematics may also be used.

There are presently active international research efforts on on all these aspects. Research groups covering the first one, besides πr^2 and non-πr^2 members of IRIF (Melliès, Ehrhard, Kesner) can be found e.g. in Birmingham (Levy, Zeilberger), Edinburgh (Plotkin), in Glasgow (McBride), Ljubliana (Simpson, Pretnar), Tallinn university (Uustalu), and within the Haskell community. For the second aspect, let us mention all groups working on the formalisation of program properties, such as the DeepSpec group in Yale (Shao), Princeton (Appel), U. Penn (Pierce, Weirich), MIT (Chlipala), Saarbrücken (Dreyer), Copenhagen (Birkedal),... The study of side effects at the junction of proofs and programs somehow stems from our team and from other members from IRIF.

At INRIA, the Focus team includes works on general notions of computation (e.g. [DLGBL17]) but with a focus on distributed systems. The Toccata team is about formal verification of program properties, but with a focus on softwares for proving properties of programs in specific programming languages. The Gallinette team is also studying effects at the junction between proofs and programs. Deducteam is studying proof theory with a focus on dependent type theory, rewriting, interoperability and proof assistants (see Section 2.7).

2.3.3 Scientific achievements

We organise the description of our achievements along the four research contexts mentioned in the positioning section.

Semantics of side effects and realisability. We include in this section works studying the general properties of side effects. In a large sense including classical realisability as an effectful kind of realisability, we also include a work on an algebraic description of classical realisability.

A theory of effects and resources. In joint work with Marcelo Fiore and Guillaume Munch, Pierre-Louis Curien considered the Curry-Howard-Lambek correspondence for effectful computation and resource management, specifically proposing polarised calculi together with presheaf-enriched adjunction models as the starting point for a comprehensive semantic theory relating logical systems, typed calculi, and categorical models in this context [55].

Interfaces for computational effects. Exequiel Rivas studied the relation between interfaces for computational effects in programming languages: arrows, idioms and monads. Building on previous results of Lindley, Yallop and Wadler, a categorical account was developed by means of monoidal adjunctions (work presented at MSFP 2018 [69] and at SYCO I).

Monads with merging. In collaboration with Mauro Jaskelioff, Exequiel Rivas developed monads with merge-like operators. These operators are based on two well-known algebraic theories for concurrency: classic process algebras and the more recent concurrent monoids (work submitted).

Relative effects: coherence for skew structures. In joint work with Mauro Jaskelioff, Tarmo Uustalu and Niccolò Veltri, Exequiel Rivas developed coherence theorems in the setting of categories with skew structures: skew monoidal categories, skew near-rig categories, skew semigroup categories. These skew structures are motivated by the study of relative effects in programming languages, where the primary example is provided by relative monads. The results are formalised in the programming language Agda. A journal article is currently being written.

Classical realisability and implicative algebras. Étienne Miquey has been working with Alexandre Miquel in Montevideo on the topic of implicative algebras. Implicative algebras are an algebraisation of the structure needed to develop a realisability model. Following the work of Munch on focalisation and classical realisability, during his postdoc in Nantes, Étienne Miquey gave alternative presentations within structures based on other connectives rather than $\rightarrow$, as was originally done by Miquel. This means disjunctive algebras (based on negation, "par") and conjunctive algebras (negation, tensor) (work formalised in Coq and presented at ITP 2018).

Certified effectful programming. Monadic constructions to reason about side effects as well as type systems to control the use of resources (which themselves can actually be seen as side effects) have been applied to programming in Haskell and Coq.

Programming with effects in Coq. Guillaume Claret defined a notion of effectful interactive computation as an embedded DSL in Coq (in the spirit of the works on algebraic effects), and used it to implement a web server. It is equipped with a dual notion of effectful interactive execution context. Using these two notions together, Guillaume Claret was able to specify and reason about interactive programs inside Coq (work presented in [74] and in his PhD manuscript).

Proof techniques for effectful programming in Coq. In collaboration with Thomas Letan (Agence Nationale pour la Sécurité des Systèmes Informatiques), Yann Régis-Gianas studied how free monads can be used to develop modular implementations and proofs of effectful systems. This proof technique is applied to the formal study of architectural attacks on IBM PC like architectures. In collaboration with Thomas Letan, Pierre Chifflier (ANSSI) and Guillaume Hiet (Centrale Supélec), Yann Régis-Gianas developed a new modular approach to model and verify effectful systems in Coq (work presented at FM 2018).

Investigations in the proof-as-program correspondence. Various investigations into relating or combining concepts from programming with concepts from logic have been carried out.

A core classical sequent calculus with dependent types. Dependent types are a key feature of type systems, typically used in the context of both richly-typed programming languages and proof assistants. Control operators, which are connected with classical logic along the proof-as-program correspondence, are known to misbehave in the presence of dependent types [Her05], unless dependencies are restricted to values. Étienne Miquey proposed a sequent calculus for classical logic and dependent types, as an extension of Curien-Herbelin's $\mu\tilde{\mu}$-calculus with a syntactical restriction to the fragment of negative-elimination free proofs of dependent types [66]. The previous calculus has been extended to arithmetic in finite types with strong elimination of existential quantification, thus providing a sequent calculus which is strong enough to prove the axiom of dependent choice [7]. The proofs of normalisation and soundness are made through a realisability interpretation of the calculus, which is obtained by using Danvy's methodology of semantic artifacts (work presented at LICS 2018, cf. Section 1).

Call-by-need. The call-by-need evaluation strategy is an evaluation strategy of the λ-calculus which evaluates arguments of functions only when needed, and then shares their evaluations across all places where the argument is needed. Étienne Miquey and Hugo Herbelin have proved the normalisation of Ariola et al.'s call-by-need λ-calculus by means of a variant of realisability whose realisers are pairs of a term and a substitution (work presented at FOSSACS 2018).

Alexis Saurin, in collaboration with Pierre-Marie Pédrot, extended their previous work on the reconstruction of call-by-need based on linear head reduction with control. They showed how linear head reduction could be adapted to the $\lambda\mu$-calculus. They showed for instance that substitution sequences of the $\lambda\mu$-calculus' linear head reduction are in correspondence with the classical Krivine abstract machine substitution sequences, validating the known fact that the KAM implements linear head reduction [67]. On top of this calculus, Kostia Chardonnet investigated levels of laziness as well as probabilistic call-by-need evaluation.

Call-by-name forcing for Dependent Type Theory Based on Pierre-Marie Pédrot's investigation of variants of the forcing construction via decomposition through call-by-push-value [10], Guilhem Jaber, Gabriel Lewertowski, Pierre-Marie Pédrot, Matthieu Sozeau, and Nicolas Tabareau studied a variant of the forcing translation for dependent type theory, moving from the call-by-value variant to a call-by-name version which naturally preserves definitional equalities, avoiding the coherence pitfalls of the former one. It allows to show various metatheoretical results in a succinct fashion, notably for the independence of axioms [61].

Gödel's functional interpretation. Pierre-Marie Pédrot gave a computational analysis of Gödel's "Dialectica" interpretation in terms of collecting stacks in an abstract machine [10]. His work also includes Dialectica for type theory, with dependent elimination.

Forcing and classical realisability. In collaboration with Boban Velickovic, Alexis Saurin advised the LMFI master internship of Ikram Cherigui on classical realisability and forcing in set theory.

The computational contents of completeness proofs. Hugo Herbelin developed an original and simple proof of Gödel's completeness theorem with side-effects which he presented in workshops.

Reverse mathematics. Reverse mathematics were studied from a provability point of view and a computational point of view, in the context of the subsystems of classical second-order arithmetic and of intuitionistic reverse mathematics of different forms of the axiom of choice, respectively.

Ramsey-type reverse mathematics. Ludovic Patey studied with Laurent Bienvenu and Paul Shafer the provability strength of Ramsey-type versions of theorems like König's lemma [13]. Ludovic Patey studied with Laurent Bienvenu the constructions of diagonal non-computable functions by probabilistic means (paper submitted). Ludovic Patey worked on the existence of universal instances in reverse mathematics [39]. He worked on the relations between diagonal non-computability and Ramsey-type theorems [40]. He studied the links between the iterative forcing framework developed by Lerman, Solomon and Towsner and the notion of preservation of hyperimmunity [46].

Reverse mathematics of Gödel's completeness theorem. Charlotte Barot, under the supervision of Hugo Herbelin, studied the relative intuitionistic strength of Gödel's completeness theorem, the ultrafilter lemma, and different forms of the Fan Theorem, as a way to transfer computational contents of proofs from one to the other theorems.

2.3.4 Collaborations

The team had a collaboration with ANSSI (Agence Nationale pour la Sécurité des Systèmes Informatiques), with Univ. of Cambridge (Fiore), Univ. of Tallinn, indirectly the Haskell development team and Univ. of Göteborg via our collaboration with Tweag I/O, Univ. de la República (Uruguay) (Miquel), Univ. of Rosario, with the Logic team (Velickovic) of the joint Math lab between Univ. Paris 7 and Univ. Paris 6, Univ. of Montpellier (Bienvenu), with Univ. of Gent (Shafer). We had the following visitors: Marcelo Fiore (Univ. of Cambridge, two weeks in February 2017), Mauro Jaskelioff (National Univ. of Rosario and CONICET, Argentina, one week in May 2018), Tarmo Uustalu (Univ. of Tallinn and Univ. of Reykjavik, two weeks in 2017 and two weeks in 2018).

2.3.5 External support

The team benefited from an invitation program to collaborate with Marcelo Fiore; from the PHC Parrot with Estonia to collaborate with Tarmo Uustalu; from the joint lab LIA IFUM with Uruguay;

from the ANR Récré (réécriture et réalisabilité). There are contacts with ANSSI for giving a formal status to our cooperation.

2.3.6 Self assessment

The theoretical research done in the team in either proof theory, program semantics, or at the junction between logic and computer science, especially regarding side effects, is original and foundational. The work on programming with side effects in Coq is also crucial to go in the direction of Coq as general-purpose programming language for certified sofware.

2.4 Research axis 2: Reasoning and programming with infinite data

2.4.1 Personnel

Permanent members: Yann Régis-Gianas, Alexis Saurin. Postdocs: Guilhem Jaber, Luc Pellissier, Andrew Polonsky. PhD students: Amina Doumane, Pierre-Marie Pédrot, Abhishek De. Interns: Paul Laforgue, Rémi Nollet, Xavier Onfroy, Sylvain Ribstein.

2.4.2 Project-team positioning

Our positioning on this topic arose as a cross fertilisation of proof theory and programming language theory in the field of logics expressing least and greatest fixed points properties and of temporal logics. Such logics allow to express (co)inductive reasoning, and the computational content is related with (co)inductive types and (co)recursive programming. Proof systems for fixed point logics can naturally be obtained by taking (co)induction rules that closely reflect fixed point theorems, e.g. Knaster-Tarski's characterisation of least fixed points as least pre-fixed points. This is the case of Kozen's famous axiomatisation [Koz83]. When building proofs using such rules, (co)invariants have to be found, possibly significantly more complex than the property to establish. An alternative to these explicit (co)induction rules is to consider proof systems featuring non-wellfounded derivation trees: this makes it possible to reason on fixed points by unfolding them, possibly infinitely often. However, the soundness of such systems requires a global *validity* condition. An important question is the comparison of such proof systems with usual systems with induction and coinduction rules à la Park. Recent results on Brotherston-Simpson's Conjecture showed that circular provability is equivalent to finitary one in presence of arithmetics for the theory of Martin-Löf's inductive definitions. However, no such general result is known for the μ-calculus. The aim of the team is to develop the (non-wellfounded) proof theory of such logics and to study their computational content via semantical studies as well as Curry-Howard correspondence.

2.4.3 Scientific achievements

This theme is part of the ANR project RAPIDO (see the National Initiatives section).

Proof Theory of Non-Well-Founded Proofs. The team contributed to the definition of new infinitary proof systems for fixed point logics where proofs can have non-wellfounded branches. Known systems were extended in three directions: new logical languages (multiplicative additive linear logic, μMALL), more flexible validity conditions (by allowing threads to bounce on axiom and cut rules) and new formal proof systems. Contributions to their meta-theory were made.

From straight to bouncing threads. After their initial work on μMALL infinitary and circular proofs establishing cut-elimination and focalisation [92], in collaboration with David Baelde and Denis Kuperberg (from ENS Cachan and Lyon), Amina Doumane, Guilhem Jaber and Alexis Saurin extended non-wellfounded proofs, by relaxing the validity condition allowing threads to move not only up in the proofs but also down by bouncing on axiom and cut inferences. Cut-elimination is proved and decidability of the criterion is investigated in the multiplicative case (work submitted) An effort in formalising in Coq circular proofs and the decidability of their validity (with straight threads) was started by Alexis Saurin together with Xavier Onfroy during the summer 2018 while Alexis Saurin and Rémi Nollet (from IRIF) started investigating the complexity of deciding validity by relating their validity to size-change termination.

Other studies on circular proof formalisms. Amina Doumane and Alexis Saurin, in a joint work with David Baelde [50], developed a game-semantics of $\mu MALL$. The definition of the bouncing validity condition helped in better understanding the relations and discrepancy between the parallel nature of threads and the sequential nature of proofs, which motivated the PhD topic of Abhishek De (advised by Alexis Saurin).

Finitising Circular Proofs. Amina Doumane proposed in her PhD a translatability criterion for finitising circular proofs. In a work with Rémi Nollet and Christine Tasson from IRIF, Alexis Saurin presented a new validity condition which can be checked at the level of elementary cycles, while the other criteria need to check a condition on every infinite branch. They still capture all circular proofs from $\mu MALL$ finite proofs [78] and obtained partial finitisation results.

Automata Theory Meets Proof Theory. In a joint work with David Baelde and Lucca Hirschi (from ENS Cachan), Amina Doumane and Alexis Saurin carried out a proof-theoretical investigation of the linear-time μ-calculus (LTμ), proposing well-structured proof systems and showing constructively their completeness for formulas expressing inclusions of Büchi automata [102]. Amina Doumane then extended the result and obtained a constructive proof of completeness for full LTμ. In order to achieve this tour de force, she identified fragments of LTμ corresponding to various classes of ω-automata and proved their completeness by using circular proof systems and finitisation of the infinite proofs in Kozen's usual axiomatisation [57] (cf. Section 1) .

Programming with fixed points, infinite data and time. In this part, we collect work on the $\lambda(Y)$-calculus, copatterns, and preliminary work on Functional Reactive Programming (FRP).

Fixpoint conjectures. With Giulio Manzonetto (from LIPN, Université Paris Nord), Andrew Polonsky (from Appalachian State Univ.) and Jakob Simonsen (from DIKU, Denmark), Alexis Saurin studied two long-standing conjectures on the λ-calculus: the "fixpoint property" and the "double-fixpoint conjecture"[1]. Their results are currently submitted to a journal [114]. Moreover, Andrew Polonsky and Alexis Saurin defined an infinitary λ-calculus allowing transfinite iteration of abstractions and ordinal sequences of applications, Λ^{o}, and established a standardisation theorem for this calculus in which the $\Lambda\mu$-calculus (as well as the Stream Hierarchy) can be embedded.

Co-Patterns. With Paul Laforgue, Yann Régis-Gianas studied the mechanisms of co-patterns from a programming language perspective, defining an untyped version as well as an abstract machine. They designed an extension of OCaml with copatterns, showed that copattern-matching can be added with no effort to any programming language equipped with second-order polymorphism and generalised algebraic datatypes (work presented at PPDP 2017 and JFLA 2018 [83]).

Functional Reactive Programming. With Sylvain Ribstein, Yann Régis-Gianas defined an OCaml library for differential FRP (DFRP) extending FRP with the ability to modify past events and compute the consequences of this modification in all the events that depend on it. Rémi Nollet studied (during his master supervised by Alexis Saurin and Christine Tasson) the recent extension of Curry-Howard correspondence between FRP and linear temporal logic (LTL) as well as various natural deduction and sequent systems for LTL and compared them to type systems for FRP.

[1] The former asserts that every λ-term admits either a unique or an infinite number of β-distinct fixpoints while the second, formulated by Statman, says that there is no fixpoint satisfying $Y\delta = Y$ for $\delta = \lambda y, x.x(yx)$.

2.4.4 Collaborations

Strong collaborations on this topics were developed with LSV (ENS Cachan) and LIP (ENS Lyon) thanks to the RAPIDO Project, as well as with LIPN (Univ. Paris 13), as well as with a number of international collaborators with the set up of the PARIS workshop on the theme of this research axis, organised during FLOC 2018 in Oxford, of which followups are being organised.

2.4.5 External support

The ANR Project RAPIDO started in January 2015 and will end in late September 2019. The project funded post-docs (Guilhem Jaber, Luc Pellissier and Andrew Polonsky) and visits, as well as the organisation of the above-mentioned PARIS workshop.

2.4.6 Self assessment

This theme is rather new in the team. Connections with the practice and development of proof assistants are not yet achieved but the proof theory of logics with induction and coinduction as well as the dynamics of those proofs have been significantly developed during the evaluation period. This opens several directions for future investigations in areas where the concept of Curry-Howard correspondence is emerging and not fully understood (for instance in the setting of FRP). One weakness is that this axis was boosted by the support of the ANR project which is coming to an end. Still, a network of collaborators has been built which could lead either to broader projects (networks and ERC Synergy grant), or/and to smaller-scale, lighter projects such a associated teams (for instance with researchers from Gothenburg Univ., Sweden).

2.5 Research axis 3: Effective higher-dimensional algebra

2.5.1 Personnel

Permanent members: Pierre-Louis Curien, Yves Guiraud, Matthieu Sozeau. Postdoc: Eric Finster. PhD students: Antoine Allioux, Cyrille Chenavier, Cédric Ho Thanh, Maxime Lucas, Jovana Obradović. Interns: Amina Bendjaafar, Joey Beauvais-Feisthauer, Jean Vassalo. External collaborators: Philippe Malbos (Univ. Lyon 1), Samuel Mimram (École Polytechnique).

2.5.2 Project-team positioning

The team was involved in the ANR Cathre project (2014-2017), coordinated by Pierre-Louis Curien, comprising most of the French researchers active in the field of the algebraic and homotopical aspects of higher categories. We actively participate to the PPS working group on higher categories, a nationally recognised seminar. In the last years, we have continued to organise international events to promote effective methods in higher algebra, to create connections with other related research directions, and to increase our international visibility: the Higher-Dimensional Rewriting and Algebra workshop in 2015, 2016, 2017 and 2018, and the Categories in Homotopy and Rewriting conference at the CIRM in Marseille in 2018.

Three current INRIA teams have close research activities. Ouragan and Polsys, at INRIA Paris, also investigate the use of rewriting-like methods, mainly based on Gröbner bases, with applications to the resolution of polynomial systems. Gallinette, at INRIA Nantes, also explores the higher structures that occur in homotopy type theory, and the formalisation of polygraphs in Coq.

2.5.3 Scientific achievements

Rewriting methods in higher algebra. With Stéphane Gaussent (Univ. Saint-Étienne) and Philippe Malbos (Univ. Lyon 1), Yves Guiraud applied rewriting methods to Artin monoids [26], improving results in representation theory: one by Tits about the presentation complex of Artin monoids [Tit81], and one by Deligne about the actions of Artin monoids on categories [Del97]. This work introduces a homotopical extension of the Knuth-Bendix completion procedure (see also [GMM13]), that was partly implemented as the Cox library and in the Rewr prototype.

With Eric Hoffbeck (Univ. Paris 13) and Philippe Malbos, Yves Guiraud has developed a theory of rewriting for associative algebras [29], with a view towards applications in homological algebra, generalising both Gröbner bases [Buc65] and Poincaré-Birkhoff-Witt bases [Pri70]. They transposed the construction of [GM12b] to compute small polygraphic resolutions of associative algebras with rewriting methods. This construction gave necessary or sufficient conditions for an algebra to have the homological property of Koszulness.

With Philippe Malbos, Yves Guiraud has written a survey (targeted for graduate students) on the use of rewriting methods in algebra, in the language of higher categories [30] Moreover, Yves Guiraud has written his "Habilitation à diriger des recherches" manuscript, on rewriting methods in higher algebra [Gui19] (defence planned is Spring 2019).

Cyrille Chenavier (PhD supervised by Yves Guiraud and Philippe Malbos [1]), has explored Berger's theory of reduction operators [Ber98] to improve noncommutative Gröbner bases. He used a lattice structure to give a new algebraic characterisation of confluence, a new completion algorithm [16], connected to Faugère's F4 procedure [Fau99], and a method to compute a linear basis of the space of syzygies of a set of reduction operations [97]. Moreover, in this setting, normalisation strategies are related through braid-like relations, leading to new constructions in homological algebra [15].

Maxime Lucas (PhD supervised by Yves Guiraud [6]) has improved the rewriting techniques of [GM12a] to obtain coherence theorems for bicategories, pseudofunctors and pseudonatural transformations [35]. He transposed to a cubical setting, and improved, the results of [GM12b]. This involved a theoretical work on the connections between globular and cubical higher categories [111], generalising several already known links in a unique setting [BH81a, BH81b, AABS02, Ste04]. He proved a cubical version of Squier's theorem [110].

With Patrick Dehornoy (Univ. Caen), Yves Guiraud has developed an axiomatic setting for monoids with a local normalisation procedure [21], generalising both normalisation by a quadratic rewriting system and Garside normalisation [DDG$^+$15]. This induced a new sufficient condition for a rewriting system to terminate, based on the length of normalisation paths on length-three words. This opened the way to improvements of rewriting methods coming from Garside theory, leading both to new methods in higher algebra, and to a new enhanced completion procedure in rewriting theory (ongoing collaboration between Yves Guiraud and Matthieu Picantin from IRIF).

Metatheoretical work on and categorification of cyclic operads. Pierre-Louis Curien and Jovana Obradović developed in [18] a syntactic approach (using some of the kit of Curien-Herbelin's duality of computation and its polarised versions by Munch and Curien) to the definition of various structures that have appeared in algebra under the names of operads, cyclic operads, dioperads, properads, modular and wheeled operads, permutads, etc. Their goal was to formalise the proofs of equivalence between various styles of definition in the literature. This was done for cyclic operads by Jovana Obradović [8]. Pierre-Louis Curien and Jovana Obradović pursued their work on cyclic operads by establishing the notion of categorified cyclic operad, where the axioms are weakened from equalities to natural isomorphisms, and by formulating conditions concerning these isomorphisms which ensure coherence [101].

As a companion to these works, in collaboration with Jelena Ivanović, Pierre-Louis Curien and Jovana Obradović have introduced an inductively defined tree notation for all the faces of polytopes arising from a simplex by truncations, and have provided many examples of its use. In their work, they also explored links between polytopes and categorified operads [36].

Opetopes. Opetopes are a formalisation of higher many-to-one operations, leading to one of the approaches for defining weak ω-categories. Opetopes were originally defined by Baez and Dolan. A reformulation (leading to a more carefully crafted definition) has been later provided by Batanin, Joyal, Kock and Mascari, based on the notion of polynomial functor. Pierre-Louis Curien, Cédric Ho Thanh and Samuel Mimram have developed (in several variants) a type-theoretical treatment of opetopes and finite opetopic sets, and have shown that the models of their type theory are indeed the opetopic sets as defined by the above authors (work submitted).

Type theory and higher algebra. Eric Finster explored the connections between intensional type theory and the theory of higher topoi, as developed in the works of Joyal and Lurie [Lur09]. In particular, in collaboration with Mathieu Anel, André Joyal and Georg Biedermann, he gave a proof of a new result about the generation of left exact modalities in higher topoi, which has a corresponding internalisation in homotopy type theory. Applications of this result to the Goodwillie calculus (an advanced technique in abstract homotopy theory) resulted in the article [11].

Antoine Allioux (PhD started in February 2018), Eric Finster, Yves Guiraud and Matthieu Sozeau are exploring the development of higher algebra in type theory based Eric Finster's internalisation of polynomial monads (of which opetopes and ∞-categories are instances) in type theory. This should provide a way to build coherent higher structures in type theory. Antoine Allioux is focusing on showing the equivalence between categories seen as polynomial monads and the standard univalent categories of homotopy type theory [Pro13]. Polynomial monads also provide a solution to the long-standing open problem of defining simplicial types in homotopy type theory.

2.5.4 Collaborations

Yves Guiraud collaborated with Patrick Dehornoy (Univ. Caen), Stéphane Gaussent (Univ. Saint-Étienne), Eric Hoffbeck (Univ. Paris 13), Philippe Malbos (Univ. Lyon 1) and Samuel Mimram (École Polytechnique), leading to four publications in international journals and to the development of the Rewr prototype. He currently works with Marcelo Fiore (Univ. Cambridge) on the first article of a planned series, and with Matthieu Picantin (Univ. Paris 7) on two articles. Pierre-Louis Curien collaborates with Bérénice Delcroix-Oger (from IRIF) on a follow-up of his work on hypergraph polytopes. Their goal is to investigate and unify various algebraic structures (known and new) on the faces of some families of polytopes. We had the following visitors: John Baez (Univ. California Riverside, one week in Nov. 2017), Marcelo Fiore (Univ. Cambridge, two weeks in Feb. 2017), Jovana Obradović (Univ. Prague, ten days in Dec. 2018).

2.5.5 External support

This axis has benefited from the support of the IDEX Sorbonne-Paris-Cité Focal project, 2013-2016 (including Cyrille Chenavier's PhD grant), the ANR Cathre project, 2014-2017 (including Maxime Lucas' PhD grant), and the ERC CoqHoTT, INRIA Gallinette, Nantes (including Eric Finster's postdoc and Antoine Allioux's PhD grant).

2.5.6 Self assessment

The team members have contributed to the development of effective methods in higher algebra, in France and abroad, leading to new results in representation theory and in homological algebra. These methods are now investigated in other places, following collaborations or dissemination of former students: Lyon, Saint-Étienne, Marseille, Nantes, Lille, Prague, etc. New potential applications have been investigated, such as the development of rewriting methods on Lie algebras, but the current known techniques are too limited to apply directly and will have to be improved (for example with the integration of Garside theory in our toolbox). Over the last years, several unforeseen theoretical obstructions have slowed down the work we have conducted (such as the complexity of the higher structures involved in rewriting theory on noncommutative polynomials), leading to a lower-than-expected software development.

2.6 Research axis 4: Incrementality

2.6.1 Personnel

Permanent members: Yann Régis-Gianas. PhD Students: Thibaut Girka, Lourdes del Carmen González Huesca. Interns: Kostia Chardonnet, Colin Gonzalez.

2.6.2 Project-team positioning

This research axis investigates a language-oriented approach to incremental computations. A computation is incremental when it consists of a sequence of small changes of its output in reaction to a sequence of small changes of its (large) input. Typically, if $f : A \to B$, then we are interested in computing the sequence $y_i = f\,x_i$ where the difference between x_i and x_{i+1} is small comparing to the size of x_i. Incremental computations are especially relevant in software development, because software is built by many patches; in machine learning, because statistical models are refined through many observations of a large dataset; or in blockchains, because the chain is built by small increments.

Our investigations in this research axis are primarily focused on the following question: what is a good language to express provable and efficient incremental computations? Around 2010, the state of the art about language-based incremental computation was the PhD thesis of Umut Acar about self-adjusting computations. In this work, Umut Acar designed a very efficient computational model to automatically incrementalise programs. Unfortunately, reasoning about the algorithmic complexity of the resulting incremental programs requires reasoning about complex dynamic mechanisms and objects (e.g. change propagations and dependency graphs of memoised computations). As a result, it is quite hard for a programmer to predict (and, by extension, to prove properties about) the efficiency of programs written in this style.

Since 2013, several teams (Klaus Ostermann in Tübingen Univ., Tiark Rompf in Purdue Univ., Conal Elliott from a data science company named *Target*, Martin Abadi from Google, etc) are trying to achieve the same kind of efficient computational models but using more declarative languages. In these differential programming languages, incrementalisation is obtained by means static program transformations so that reasoning about an incremental program is a standard proof of program. Our work is focusing on differential higher-order functional programs computing over non-numerical domains while most of the aforementioned teams are implementing efficient declarative programming models to automatically differentiate first-order numerical programs. On the topic of λ-terms' static differentiation, we collaborated with Klaus Ostermann's team.

One interesting application of differential programming is the specification and certification of software evolutions, as initiated by Thibaut Girka, whose PhD thesis was funded by Mitsubishi Electrics. By constructing a (non-numeric) domain for program semantic differences, we provide a foundational framework to perform relational reasoning about the difference of behaviour between programs. The verification of software evolutions is an important topic in the verification community, with strong contributions by teams from Microsoft Research, IMDEA, Univ. of Technion, and IRIF. Comparing to these approaches which prioritise automation, we focus on expressivity: for instance, we are able to reason about bugfixes while other approaches of the literature cannot.

2.6.3 Scientific achievements

Differential programming language. Lourdes González and Yann Régis-Gianas developed a new variant of the differential λ-calculus that has two main features: (i) it is deterministic; (ii) it is based on a notion of first-class changes. An extension of the simply-typed λ-calculus with differentials and partial derivatives offers a language to reason about incrementality. The basic system, λ-diff, was enriched with expressions for fixpoints and data-types together with their corresponding derivatives to analyse incrementality over them. The above results are reported in the second part of Lourdes González' PhD thesis [5].

In collaboration with Paolo Giarrusso, Philipp Schuster and Yufei Cai (Univ. Marburg, Allemagne), Yann Régis-Gianas developed a new method to statically incrementalise higher-order programs using formal derivatives and static caching. This method overcomes a defect of previous approaches to static differentiation of programs by sharing information between base computations and incrementalised computations. He developed a mechanised proof for this transformation, as well as a prototype language featuring efficient derivatives for functional programs.

Differential functional programming is new, and some exploratory work is needed to understand what are its idioms. In collaboration with Olivier Martinot (Univ. Paris 7), Yann Régis-Gianas studied a new technique to implement incrementalised operations on lists. In collaboration with Colin Gonzalez, Yann Régis-Gianas developed BLACS, a programming framework that applies differential functional programming techniques to the implementation of asynchronous spreadsheets for big data. In collaboration with Lelio Brun (ENS), Yann Régis-Gianas developed DeltaCoq, a library for certified incremental functional programming.

Difference languages. In collaboration with David Mentré (Mitsubishi), Thibaut Girka and Yann Régis-Gianas designed a new algorithm for correlating program generation: such a program is used to characterise the differences between two close programs. Before their work, only one, unsound. algorithm existed in the literature. The new algorithm is sound and certified in Coq [58]. They also designed a metatheoretical framework to develop verifiable difference languages in Coq. Such formal differences capture semantic differences between close programs and overcome existing systems in terms of expressivity [59].

Kostia Chardonnet and Yann Régis-Gianas started the formalisation of difference languages for Java, using the framework developed by Thibaut Girka. In particular, Kostia Chardonnet implemented a mechanised small-step operational semantics for a large subset of Java.

2.6.4 Collaborations

Yann Régis-Gianas collaborates with David Mentré (Mitsubishi Electrics, Rennes), with Paolo Giarrusso (EPFL, Suisse), with Philip Schuster and Yufei Cai (Univ Marburg, Allemagne). We received the following visitors: Paolo Giarrusso (Univ. of Marburg, February 2016), Lourdes González (Univ. of Mexico, December 2016).

2.6.5 External support

The PhD thesis of Thibaut Girka was supported by Mitsubishi (CIFRE grant). The PhD thesis of Lourdes González was funded by the ANR project Paral-ITP.

2.6.6 Self assessment

The work done on differential λ-calculus is foundational: it provides an important robust basis to build further research about incremental computations. It is however a pity that the results of Lourdes González' PhD thesis about differential λ-calculus have not been published quickly enough: almost the same results have been published at the same period by the team of Klaus Ostermann. Fortunately, the competition with that team finally led to a collaboration. Incremental programming is notoriously difficult: our ability to certify incremental programs using Coq is a strength that will be emphasised in the next years.

The theoretical framework of the PhD thesis of Thibaut Girka has now to be scaled up to realistic programming languages that interest the software engineering community. The collaboration with Mitsubishi is continuing in that direction. A new PhD thesis is planned to that end.

2.7 Research axis 5: Metatheory and development of Coq

2.7.1 Personnel

Permanent members: Hugo Herbelin, Pierre Letouzey, Yann Régis-Gianas, Matthieu Sozeau. PhD students: Gaëtan Gilbert (Inria Nantes), Gabriel Lewertowski, Cyprien Mangin, Théo Winterhalter (Inria Nantes), Théo Zimmermann. Engineer: Thierry Martinez. Interns: Meven Bertrand, Philipp Haselwarter, Vadim Zaliva. External Collaborators: Nicolas Tabareau (Inria Nantes), Beta Ziliani (Córdoba, Argentina).

2.7.2 Project-team positioning

This line of research centers around the study of dependent type theories and their implementation, notably in the Coq proof assistant. Within Inria, the Gallinette team, with which we have a strong collaboration, and the Deducteam team are the closest. Internationally, dependent type theory is a very active area of research, especially since the advent of homotopy type theory, with many groups strongly focusing on it: Gothenburg (Thierry Coquand), Nijmegen (Herman Geuvers), Nottingham (Thorsten Altenkirch), Strathclyde (Neil Ghani), Aarhus (Lars Birkedal), Birmingham (Martin Escardó), CMU (Steve Awodey, Robert Harper), Microsoft (Leo de Moura). While many of these research groups concentrate on the semantical side, our focus is more on the implementation side of proof assistants, for applications to both programming and mathematics.

The πr^2team is at the forefront of the development of Coq. On the theoretical side, the team has developed new extensions of the proof assistant that put it at the state of the art in the treatment of universes and cumulativity in type theory-based proof assistants (Agda, Lean, RedPRL), along with formalising the unification algorithm of the system whose interaction with universes is non-trivial. A recent development coming after many years of maturation is the integration of definitional proof-irrelevance in the system, which was developed in collaboration with the Gallinette team and a member of the Agda development team. The system is also gaining traction for the development of dependently-typed programs through the EQUATIONS plugin which provides state-of-the-art compilation of dependent pattern-matching and tools to reason on functional programs.

Development of Coq itself has become more decentralised and is a collaborative effort managed mainly by the πr^2, Gallinette, Tocatta and Marelle teams, with Matthieu Sozeau serving as project coordinator, after Hugo Herbelin. More information about this can be found in Section 3.

2.7.3 Scientific achievements

Cumulativity for Inductive Types. Together with Amin Timany, Matthieu Sozeau developed the Calculus of Cumulative Inductive Constructions, which extends the cumulativity relation of universes to universe polymorphic inductive types [70]. Its consistency model [90] suggested a suprising relaxation of the subtyping relation on inductive types in Coq, able to model the previously ad-hoc treatment of polymorphism for inductive types. This work provides a state-of-the-art universe system for Coq, with applications to formalisations of homotopy type theory like is done in the HoTT library [52], category theory and syntactic translations of type theory [BPT17].

Proof irrelevance and homotopy type theory. During his master's internship supervised by Matthieu Sozeau, Philipp Haselwarter studied a formulation of proof-irrelevance based on the decomposed presentation of CIC by Spiwack and Herbelin [HS13]. Following this work, Gaëtan Gilbert, Nicolas Tabareau and Matthieu Sozeau developed the theory and implementation of *strict* or definitionally proof-irrelevant propositions in the Calculus of Inductive Constructions. In collaboration with Jesper Cockx (Chalmers), they developed this notion in full in an article presented at POPL 2019 [27].

Dependent pattern-matching and recursion. Cyprien Mangin and Matthieu Sozeau developed a complete rewriting of the EQUATIONS plugin which provides a high-level interface for the definition of mutual, nested and well-founded recursive definitions using dependent pattern-matching in Coq, without axioms. The tool also derives useful reasoning principles on these definitions, significantly raising the scope of definitions handled by previous tools (FUNCTION and PROGRAM), and turning Coq into a viable alternative to its cousins Agda and Idris for dependently-typed programming. Developing this new version of EQUATIONS was a significant engineering effort that spanned the whole evaluation period. An earlier version of the system was presented at LFMTP 2015 [64] and an article describing the latest version is in revision [112].

An alternative simplification of dependent pattern-matching avoiding the use of equalities and resulting in smaller proof terms is being implemented by Thierry Martinez, following the PhD thesis work of Pierre Boutillier and the internship work of Meven Bertrand. The algorithm based on small inversion and generalisation is the object of a paper to be submitted to the TYPES post-proceedings.

Unification. Matthieu Sozeau worked in collaboration with Beta Ziliani (PhD at MPI-Saarbrücken, now assistant professor at Córdoba, Argentina) on formalising the unification algorithm used in Coq, a central component of type-checking and tactical reasoning in the proof assistant, with which the user interacts all the time. This resulted in a precise formalisation of all the rules of unification including the ones used for canonical structure resolution and universes, presented in a manner accessible to users (work presented at ICFP 2015 [72] and refined into a journal article [48]).

Certified compilation and meta-programming. MetaCoq[2] is a project led by Matthieu Sozeau, in collaboration with Simon Boulier and Nicolas Tabareau in Nantes, Abhishek Anand and Gregory Malecha (BedRock Systems, Inc) and Yannick Forster in Saarbrucken. The project extends the Template-Coq reification plugin of Malecha with specifications of the typing rules and basic metatheoretical properties of the system, a reference type checker for Coq, written in Gallina, an implementation of extraction and a monad for programming deeply embedded plugins in Coq itself, in the style of MTac. The foundation of this project was presented at ITP 2018 [49], and a journal article is in preparation.

Matthieu Sozeau participates to the CertiCoq project[3] led by Andrew Appel at Princeton Univ., whose aim is to verify a compiler from Coq's Gallina language down to CompCert C-light which provides itself a certified compilation path to assembly language. Matthieu Sozeau authored the front-end part of CertiCoq including extraction, integrated in MetaCoq, along with another phase. The CertiCoq team expects to release a first version of the compiler in the beginning of 2019, along with an article describing it.

In collaboration with Jan-Oliver Kaiser (MPI-SWS), Beta Ziliani (CONICET/FAMAF), Robbert Krebbers (ICIS) and Derek Dreyer (MPI-SWS), Yann Régis-Gianas participates in the Mtac2 project, a shallowly embedded metaprogramming language for Coq. The new version of this language has been presented at ICFP 2018 [33]. It includes in particular a dependently-typed variant of the LCF tactic typing discipline. In collaboration with Xavier Denis (Univ. Paris 7), Yann Régis-Gianas is implementing a compiler for Mtac2.

Extensionality and Intensionality in Type Theory. Théo Winterhalter, Nicolas Tabareau and Matthieu Sozeau studied and formalised a complete translation from Extensional to Intensional Type Theory in Coq, that was presented at CPP 2019 [71]. Using the MetaCoq framework, this establishes formally that only the Functional Extensionality and Uniqueness of Identity Proofs

[2] http://metacoq.github.io/metacoq
[3] https://www.cs.princeton.edu/~appel/certicoq

principles are necessary to effectively embedd ETT in ITT.

Equivalences for free!. Nicolas Tabareau (Inria Nantes), Eric Tanter (Univ. Chile in Santiago) and Matthieu Sozeau developed a new parametricity translation for justifying the transport of programs and proofs by equivalences in type theory (work presented at ICFP 2018 [47], see Section 1).

Development of Coq. The amount of contributions to the Coq system increased significantly in the recent years. Hugo Herbelin, Matthieu Sozeau and Théo Zimmermann, helped by members from Gallinette (Nantes) and Marelle (Sophia-Antipolis), devoted an important part of their time to coordinate the development, to review propositions of extensions of Coq from external and/or young contributors, and to propose themselves extensions. During the evaluation period, versions 8.5 through 8.9 of Coq were released (see Section 3 for more details).

Software engineering aspects of the development of Coq. Théo Zimmermann has studied software engineering and open collaboration aspects of the development of Coq. In collaboration with Annalí Casanueva Artís from the Paris School of Economics, he studied the migration of the Coq bug tracker from BugZilla to github. The results show an increased number of bugs by core developers and an increased diversity of the people commenting bug reports, and validate *a posteriori* the usefulness of such a switch. A paper [117] was presented at the EAQSE workshop.

Théo Zimmermann has founded the coq-community GitHub organisation in July 2018. This is a project for a collaborative, community-driven effort for the long-term maintenance and advertisement of Coq packages. Théo Zimmermann and Yann Régis-Gianas are preparing an article of the model proposed by the various existing *-community GitHub organisations.

2.7.4 Collaborations

Matthieu Sozeau is a participant in the ERC CoqHoTT led by Nicolas Tabareau and the CertiCoq project led by Andrew Appel at Princeton Univ. since 2015. Andrej Bauer (Univ. of Ljubljana) visited πr^2 and PPS for one month in September 2015 to collaborate with Matthieu Sozeau and Philipp Haselwarter on the subject of proof irrelevance and squash types. Phillip Haselwarter went on to do a PhD with Bauer. Beta Ziliani (Univ. of Córdoba) is a long time collaborator of both Matthieu Sozeau and Yann-Régis Gianas on the topics of unification (ICFP 2015, JFP 2016) and tactic languages (ICFP 2018). Amin Timany (KU Leuven, Belgium) visited the team for two months in March-April 2017 and collaborated with Matthieu Sozeau on the design and implementation of cumulative inductive types in Coq throughout 2017-2018. Vadim Zaliva (PhD student at CMU) visited the team for one month in July 2018 and collaborated with Matthieu Sozeau on the use of Template-Coq to verify translations from shallow to deep embeddings (publication at CoqPL 2019).

2.7.5 External support

The work on this axis was partly supported by the ERC CoqHoTT. Coq development was supported by an Inria ADT and specific support from Inria's DGD-T.

2.7.6 Self assessment

The development of the core type theory of Coq continued with the addition of a more sophisticated universe system and the introduction of strict propositions, which clearly benefit to users. However, we are currently lacking ways to ensure without a doubt the consistency of these extensions and the safety of the implementation. We should prioritise the verification of the implementation of Coq's kernel and the metatheoretical study of its type system using MetaCoq.

Unfortunately, two PhD students preferred to leave for industry before completing their thesis. In a somewhat related vein, the increasing demanding activity on Coq is quite time-consuming for some members of the team, despite the improvement of the engineering support for Coq.

2.8 Miscellaneous results

This subsection lists the results achieved by the team in other directions than the five research axes.

Proofs and surfaces. Following ideas of Richter-Gebert, Pierre-Louis Curien, together with Jovana Obradović (while on her postdoc in Prague), joined a project with Zoran Petrić and other Serbian colleagues on formalising proofs of incidence theorems (arising by repeated use of Menelaus theorem) by means of a cyclic sequent calculus, which has been proved sound and been experimented on an extended set of examples from elementary projective geometry. A paper is being written.

Hofstadter nested recursive functions and Coq. Pierre Letouzey has been studying a family of nested recursive functions proposed by Hofstadter, see https://oeis.org/A005206 and https://oeis.org/A123070. Some new results have been proved in Coq and presented in [89]. A generalisation of this work is ongoing, see https://www.irif.fr/~letouzey/hofstadter_g/. Many points remain to be explored, like the surprising occurrence of a Rauzy fractal during the investigations, see https://www.irif.fr/_media/rencontres/pps2018/letouzey.pdf.

Real Numbers in Coq. The present Coq library of real numbers is made of 17 axioms. Daniel de Rauglaudre has been studying the possibility of making an implementation with one only axiom: the Limited Principle of Omniscience (LPO), which says that we can differentiate an infinite sequence of 0s from an infinite sequence holding something else than 0 This axiom had been already used in the formal proof of Puiseux' theorem done some years ago (only axiom of this proof too). Real numbers are defined by an infinite sequence of digits and the operations of addition and multiplication by algorithms using LPO. It was tested in OCaml, the axiom being replaced by a function having a limit corresponding to the precision of the computation and it seems to work. But the proof in Coq that this implementation is a field stumbles on difficulties about the associativity of addition which is more complicated to establish than expected.

Proofs of algorithms on graphs. Jean-Jacques Lévy and Chen Ran (PhD student, Institute of Software, Beijing) pursue their work about formal proofs of graph algorithms. In 2016, they completed proofs for algorithms computing the strongly connected components in graphs (Kosaraju 1978 and Tarjan 1972). This work was done using the Why3 system (team Toccata, Saclay) and the numerous automatic provers interfaced with Why3. A very minor part of these proofs is also achieved in Coq. The difficulty of this approach is to combine automatic provers and the intuitive design, and to find the good level of abstraction in order to avoid too many implementation features while keeping an effective presentation. In 2017, the same proofs were fully completed in Coq-SSReflect with the Mathematical Components library by Cohen and Théry (team Marelle, Sophia-Antipolis), and in Isabelle-HOL by Merz (team VeriDis, Nancy), both proofs with the assistance of Jean-Jacques Lévy. These proofs are between a factor 3 to 8 in length with respect to the initial Why3 proofs On the way, this collaboration led to a new, better presentation of the Why3 proof. These works were presented at JFLA 2017, at VSTTE 2017 in Heidelberg, and a paper is under submission. Scripts of proofs can be found at http://jeanjacqueslevy.net/why3, where other proofs of graph algorithms are also presented: acyclicity test, articulation points, biconnected components. A proof of Tarjan's planarity test is also under design.

Detecting k-synchronisability violations. As part of the research program of the Inria-CAS project VIP, Ahmed Bouajjani (Univ. Paris 7), Constantin Enea (Univ. Paris 7), Kailiang Ji and Shaz Qadeer (Microsoft Research) programs gave a procedure for deciding k-*synchronisability* of a program, i.e. whether every computation is equivalent (has the same happens-before relation) to one of its k-synchronous computations. They also showed that reachability over k-synchronous computations and checking k-synchronisability are both PSPACE-complete [86].

2.9 Evolution of research directions during the evaluation period

We comment on the objectives planned in the 2015 evaluation report of the team.

2.9.1 Effects in proof theory and programming

Effects in proof theory. Beside the question of the strength of classical realisability, we addressed all our objectives, though, unfortunately, in large part due to the increasing activity on Coq, without always having enough time to concentrate on finalising a paper out of our results.

Lazy evaluation and its dual. Theoretical studies say that there is a dual to lazy evaluation and it remains interesting to evaluate whether there exist effective incarnations of it in the wild. We however did not make progresses on this question.

Game semantics of lazy evaluation. We did not explore game semantics of lazy evaluation, which was a relatively marginal point in our project.

2.9.2 Reasoning and programming with infinite data

Infinite proofs, fixed point logics and coinductive reasoning: syntax and semantics. In this direction, the efforts were put on the development of a syntax and semantics for fixed-points and infinitary logics, where the results we obtained go far beyond our original expectations. With respect to copatterns and coequation, the dependently-typed case was not addressed, but some implementation was developed instead as reported above.

Automata theory meets proof-theory. Original plans were to develop a framework for non-determinism in Ludics and to lift the existing framework for deterministic finite automata and formal languages in computational ludics to non-deterministic finite automata and ω-automata. Some investigations in the early phase of Amina Doumane's PhD showed that the problem was much more complex than expected (both in the treatment of existential non-determinism and in expressing the ω-acceptance condition in a way that is compatible with the technical setting under consideration). The research effort in connecting automata theory and proof-theory was therefore reoriented towards the development of the non-wellfounded proof-theory of the linear-time μ-calculus and the refinement of classical relationships between LTμ-calculus and automata, in order to design a new, constructive proof of completeness of the μ-calculus.

Functional reactive programming. While proof systems for LTL and type systems for FRP were indeed investigated in Rémi Nollet's Master and early PhD work and in Amina Doumane's PhD, the denotational semantics of FRP has not yet been addressed and is still in plans for future works.

2.9.3 Effective higher-dimensional algebra

Algebraic methods for the word problem. The improvement of the Knuth-Bendix completion started with [21], which established a theoretical setting to transfer methods from Garside theory to rewriting; this led to the ongoing development of the new KGB completion procedure, enhanced with elements of Garside theory. We planned a generalisation of the connexion between convergent presentations and the word problem for monoids, based on [OKK98]; we supervised two M2 interns on this theme, but the students did not pursue in PhD.

Improvement of the homotopical completion-reduction procedure. An unforeseen hitch arose while developing rewriting theory in a linear setting, and took much of our energy: indeed, the article [29] on rewriting methods for associative algebras was considered finished at the previous evaluation, but contained a gap in a proof that finally induced three more years of work.

Cubical polygraphic resolutions were studied by the PhD thesis of Maxime Lucas [6], who went way further than the initial expectations in the understanding of the cubical setting and the development of effective computational methods.

The link between confluence monoids and resolutions was investigated by the PhD thesis of Cyrille Chenavier [1], leading to more effective methods than planned; he also worked on several other related themes, mostly around an algebraic account of completion.

The planned work on Steiner theory was quickly cancelled, after a theoretical obstruction was found. The investigation on higher homotopical reduction was harder than expected, but an article containing the first results in this direction is almost finished.

Weak higher-dimensional categories. We made mild progress on the coherence of weak higher categories during the PhD thesis of Maxime Lucas [6], who studied the coherence of weak natural transformations; an ongoing collaboration with Marcelo Fiore is expected to provide new insights thanks to the point of view of n-oidal categories. The interaction with homotopy type theory is the main subject of the recently started PhD thesis of Antoine Allioux, although with a different point of view than planned, where opetopes replace cubes as the basic blocks of higher structures. This line of work should connect with the syntactic work on opetopic sets reported above.

2.9.4 Incrementality

Incremental computing based on differential functional programming. We consider the computational model for differential functional programming to be stabilised now and hence this objective to be fulfilled.

Certifying incremental computation. The cache-transfer-style static differentiation has been proved correct in Coq. The next challenge is to equip this functional programming language with a proof system to be able to prove properties about incrementalised programs, especially about their execution costs and their asymptotical complexity.

Differential semantics. As a result of Girka's PhD thesis, we already have a usable formal framework to reason about semantic differences of programs. We want to know if it can scale up to more realistic programming languages.

Certified incremental proof-checking. There is no significant progress on this objective, because we focused on certifying incremental programs. This direction is simply postponed.

2.9.5 Metatheory and development of Coq

Monadic effectful programming in Coq. Our study of Coq extensions using monadic approaches has started with the work about simulable monads and certified interactive computations. Unexpectedly, it has continued with Mtac2 and with FreeSpec. Because we dedicated our time to these last two important results, we postponed our specific plan to look for more efficient computational models to execute programs based on simulable monads. This investigation is simply postponed and could actually benefit from recent advances about algebraic effects to enable a closer interaction between efficient untrusted oracles and less efficient certified validators.

Typed tactics. With Mtac2, we consider that the objective to legitimate Mtac as a credible alternative to Ltac is achieved. The goal to get efficient tactics by means of compilation is an active research project and should lead to a practical implementation very soon. A last goal we had in mind was to explore domain specific proof script languages. This has not yet been considered, essentially by lack of time.

Equality in type theory. The objective of relating typed and untyped intensional equality in the presence of subtyping did not see progress during the evaluation period, as we focused on other items.

The work on strict propositions [27] directly addresses the objective about proof-irrelevance in Coq, providing a definitional proof-irrelevant sort in Coq.

An ongoing work of Antoine Allioux, Eric Finster, Yves Guiraud and Matthieu Sozeau on the definition of coherent higher-dimensional structures in type theory addresses the main obstacle in the objective of understanding coherent structures in type theory.

In the direction of capturing fragments of extensional equality, the work of Théo Winterhalter, Matthieu Sozeau and Nicolas Tabareau [71] on an effective translation from extensional to intensional type theory allows for the inclusion of extensional equality in type theory, by translation. In the course of this investigation, it appeared that restricting the reflection rule to particular types for which UIP is provable is not enough to prevent UIP at every type, making the idea of restricting extensionality to a fragment (in a 1-level type theory) dubious.

Our work on formalising the unification algorithm of Coq is an important step in providing a formal underpinning to this central part of the proof assistant. We plan to pursue the second part of mechanising the unification algorithm in Coq as part of the MetaCoq project (new objective).

The new version of EQUATIONS addresses the objective of permitting definitions by pattern-matching using definitional or propositional versions of the UIP principle on the concerned types. The tool was also greatly generalised to handle complex recursion schemes (new objective).

Coq as a general-purpose platform for certified proofs. The main objective here was to open up the development process and modularise the architecture of the system, so that it can be more easily maintained and extended. As described previously in this report, the development model of Coq substantially changed during the evaluation period, allowing more developers to contribute and work concurrently on the system. While our objective of modularisation is not yet attained, a number of efforts go into this direction: functionalisation and isolation of interfaces and components, deprecation of redundant components, and, in general, higher requirements on the quality of the code being integrated. The management, development and maintenance of software of this size, with around a dozen core developers and a growing number of external contributors, is a huge effort. We are thankful for Inria's help in recruiting two engineers through the Coq consortium that are focusing on these tasks. Nevertheless, the overall increase of the programming and coordinating activity on Coq took up a lot of the time of the members of the team involved in the development.

3 Knowledge dissemination

3.1 Publications

	2015	2016	2017	2018+
PhD Theses	2	2	4	2
H.D.R. (*)				1 (Spring 2019)
Journals	6	7	15	9
Conference proceedings (**)	8	3	6	7
Book chapters	1			1
Technical reports	1	1	1	

(*) Habilitation à diriger des recherches

(**) Conferences with a program committee

The most important journals for the team are (number of accepted papers in parentheses):

- Journals in theoretical computer science or logic, such as: Journal of Formalized Reasoning (1), Journal of Functional Programming (1), Journal of Symbolic Computation (1), Journal of Symbolic Logic (1), Logical Methods in Computer Science (1), Mathematical Structures in Computer Science (3), Proceedings of the ACM on Programming Languages (3).
- Mathematical journals with a general audience, such as: Advances in Mathematics (2), Compositio Mathematica (1), Mathematische Zeitschrift (1).
- Mathematical journals specialised in algebra and topology, such as: Algebras and Representation Theory (1), International Journal of Algebra and Computation (1), Journal of Homotopy and Related Structures (1), Journal of Pure and Applied Algebra (1), Journal of Topology (1).

The top two conferences in our themes are Principles of Programmming Languages (1) and Logic in Computer Science (2). Other important conferences include: Certified Programs and Proofs (2), Computer Science Logic (1), European Symposium on Programming (2), Formal Structures for Computation and Deduction (2), Foundations of Software Science and Computation Structures (2), Interactive Theorem Proving (1), International Conference on Functional Programming (1) Principles and Practice of Declarative Programming (2).

3.2 Software

Coq. An interactive proof assistant based on type theory. Web site: `http://coq.inria.fr`. Self-assessment:

- Audience: A-5 (wide audience, large user's community).
- Software originality: SO-4 (original software implementing a fair number of original ideas).
- Software maturity: SM-4 (major software project, strong software engineering).
- Evolution and maintenance: EM-4 (well-defined and implemented plans for future maintenance and evolution).
- Software distribution and licensing: SDL-4/5 (public source or binary distribution on the Web). Binary packages available for common platforms.

Coq 8.5 final was relased in January 2016, followed by 8.6 (December 2016), 8.7 (October 2017), 8.8 (April 2018) and 8.9 (February 2019). Among many fixes and improvements, these versions bring:

- Better performance and stability, along with cleanups of the interfaces of the system.
- Stabilisation and improvements of asynchronous document processing and compilation, universe polymorphism, the new proof engine and many tactics.
- Integration of the SSReflect plugin providing a proof language centered on small-scale reflection.
- Better tool support for deprecation, control of warnings, and building of contributions and plugins.
- A collaborative migration of the documentation to the Sphinx tool allowing easier community improvements.

During these four years, the development of Coq moved to an entirely decentralised, collaborative model. We set up procedures for contributing and a formal review process along with a move to modern software engineering tools: using github for code, pull requests, reviews and issues, and setting up a thorough continuous integration system. It attracts now dozens of external contributors and has a fast-paced development cycle. As such, Coq is probably the most widely used and developed proof assistant on the market. Coq is gradually moving towards an industry-strength tool, compared to its direct competitors Lean and Agda, as attests its continued use for large verification projects. The NSF DeepSpec Expedition project is such an example: its aim is to formalise and connect specifications of a full stack from hardware to OS to user-level software in Coq.

EQUATIONS. A plugin for Coq elaborating high-level recursive function definitions using dependent pattern-matching to Coq's kernel type theory. Web site: http://mattam82.github.io/Coq-Equations/. License: LGPL 2.1. Self-assessment:

- Audience: A-4 (large audience, used by people outside the team).
- Software originality: SO-4 (original software implementing a fair number of original ideas).
- Software maturity: SM-3 (well-developed software, good documentation, reasonable software engineering).
- Evolution and maintenance: EM-3 (good quality middle-term maintenance).
- Software distribution and licensing: SDL-4 (public source or binary distribution on the Web).

EQUATIONS competitors are the Agda and Idris dependently-typed programming languages which provide similar functionality albeit without the verification guarantees of an elaboration, and the Function definition packages of Lean and Isabelle/HOL. The first production-ready version was released in December 2017 and initial feedback from users is encouraging.

Rewr. A prototype of computer algebra system, using rewriting methods to compute resolutions and homotopical invariants of monoids. The library implements various classical constructions of rewriting theory (such as completion), improved by experimental features coming from Garside theory, and allows higher algebra computations based on Squier theory. Specific functionalities have been developed for usual classes of monoids, such as Artin monoids and plactic monoids. Web site: http://www.lix.polytechnique.fr/Labo/Samuel.Mimram/rewr. Self-assessment:

- Audience: A-3 (ambitious software, usable by people outside the team).
- Software originality: SO-4 (original software implementing a fair number of original ideas).
- Software maturity: SM-2 (basic usage works, terse documentation).
- Evolution and maintenance: EM-3 (good quality middle-term maintenance).
- Software distribution and licensing: SDL-4 (public source or binary distribution on the Web).

Catex. A Latex package and an external tool to typeset string diagrams easily from their algebraic expression. Catex works similarly to Bibtex. Web site: https://www.irif.fr/~guiraud/catex/catex.zip. Self-assessment:

- Audience: A-2 (used by people in the team or close to the team).
- Software originality: SO-4 (original software implementing a fair number of original ideas).
- Software maturity: SM-3 (well-developed software, good documentation, reasonable software engineering).
- Evolution and maintenance: EM-3 (good quality middle-term maintenance).
- Software distribution and licensing: SDL-4 (public source or binary distribution on the Web).

Cox. A Python library for the computation of coherent presentations of Artin monoids, with experimental features to compute the first dimensions of the Salvetti complex. Web site: https://www.irif.fr/~guiraud/cox/cox.zip. Self-assessment:

- Audience: A-1 (internal prototype).
- Software originality: SO-3 (original software reusing known ideas and introducing new ideas).
- Software maturity: SM-2 (basic usage works, terse documentation).
- Evolution and maintenance: EM-1 (no real future plans).
- Software distribution and licensing: SDL-4 (public source or binary distribution on the Web).

3.3 Technology transfer and socio-economic impact

The team did not undertake a sustained action of transfer, however the impact of Coq in industry is starting to show and we have informal contacts with industry users. In particular, we know of uses of Coq at Intel (Michael Soegtrop is a regular Coq contributor), Tweag I/O (Arnaud Spiwack is a close collaborator), Mitsubishi, Facebook, Google (collaborators of Adam Chlipala's group at MIT), the FireEye company (which hired a team of Coq proof engineers on a cyber-security project, now cancelled), BedRock Systems Inc. (which employs Gregory Malecha, a collaborator of Sozeau), and Edukera (a French company whose goal is to bring proof assistants to the classroom), among others. Our hope is that the Coq consortium, a structure meant to organise the Coq community of users from academia and industry can ultimately undertake potential technology transfer activities. Its setup, pushed mainly by Yves Bertot and Maxime Dénès in the Marelle team, has however encountered administrative difficulties until now. Due to the completely open source nature of Coq, however, this did not actually prevent industrial uses of the system.

3.4 Teaching

Research master courses (from 10 to 30 hours per course). Pierre-Louis Curien taught every year in the MPRI Master 2 course "Models of programming languages: domains, categories, games" (together with Thomas Ehrhard and Paul-André Melliès). He taught courses on homotopy type theory, and on the Foundations of Programming Languages at East China Normal University (ECNU), Shanghai (June 2017 and November 2018).

Yves Guiraud taught a course on the applications of rewriting methods in algebra in the Master 2 Mathématiques Fondamentales of Lyon (January 2017).

Hugo Herbelin taught a class at the LMFI Master 2, entitled "Preuves et programmes en logique classique" in 2015, 2016 and 2017 (48h each year) and on homotopy type theory in 2018 (24h, collaboration with Nicolas Tabareau).

Since 2015, Pierre Letouzey has taught two courses to the LMFI Master 2 students: "Introduction to computer-aided formal proofs", and an introduction to programming, initially in OCaml, now in Coq (48h per year).

Pierre Letouzey and Matthieu Sozeau were lecturers at the "École de Printemps d'Informatique Théorique" on "Preuves mécanisées de programmes" in Fréjus in May 2015.

Yann Régis-Gianas took part every year in the MPRI Master 2 course entitled "Type systems": he gave lectures about generalised algebraic data types, higher-order Hoare logic and dependently-typed programming.

Alexis Saurin taught, jointly with Christine Tasson, an LMFI Master 2 course entitled "Outils classiques pour la correspondance preuves-programmes" in 2015-2016 and in 2018-2019. In 2016-2017 and 2017-2018, he taught the Master 2 course entitled "Cours fondamental de logique: théorie de la démonstration". He chaired the LMFI Master 2 from 2013 to 2018.

Matthieu Sozeau taught the MPRI Master 2 course on "Advanced uses of proof assistants" for the whole 2015-2019 period with Bruno Barras (Inria Deducteam), 12h/year. He gave a guest lecture on dependent pattern-matching and EQUATIONS at the Univ. of Saarland in April 2018, and an introductory lecture on dependent type theory at the EUTYPES summer school in Ohrid, Macedonia, in August 2018. He taught a course at the EJCP 2017 summer school in Toulouse in June 2017 and at EJCP 2018 in Lyon in July 2018, on an introduction to interactive theorem proving.

Other teaching. Pierre Letouzey's regular duty as teacher in the Computer Science department of Univ. Paris 7 amounts to 192h per year. Since 2015, Pierre Letouzey taught in particular "Compilation" to Master 2-Pro students (50h/year) and "Computed-aided formal proofs" to Master 1 students (36h/year).

Yann Régis-Gianas performed the same amount of teaching hours (except in 2018 and 2019 because he obtained a part-time sabbatical funded by INRIA). He is usually responsible for the Master 1 course "Compilation" (50h/year), for the Master 2 Pro course "Comparative programming" (50h/year) and for the Master 2-Pro course "Advanced Concepts in Object Oriented Programming" (36h/year).

Internship supervision. Pierre-Louis Curien supervised Akira Yoshimizu, who had a six-month INRIA international internship (Nov. 2014 - April 2015), and worked on abstract machines for the geometry of synchronisation,

Yves Guiraud supervised the Master 2 internships of Pierre Giraud (M2 LMFI, Univ. Paris 7, 2015), Amina Bendjaafar (M2 LMFI, Univ. Paris 7, 2016) and Jean Vassalo (École Polytechnique, 2018), and a 6-month research internship of Joey Beauvais-Feisthauer (L3 Univ. Ottowa, 2016).

Hugo Herbelin supervised the L3 internship of Meven Bertrand and the pre-doctoral internship of Théo Zimmermann in 2016, as well as the M2 internship of Charlotte Barot in 2017.

In 2015, Yann Régis-Gianas supervised the M2 internships of Béatrice Carré and Jacques Pascal Deplaix, and the M1 internships of Lélio Brun and of Loïc Runarvot. In 2016, he supervised the M1 internships of Paul Laforgue and Sylvain Ribstein, the L3 internship of Kostia Chardonnet, and the M2 internship of Colin Gonzalez. In 2018, he supervised Loïc Peyrot (Master 1, Paris-7) about the development of a tool to define exercises for the learn-ocaml platform in a single ML file; Carine Morel (Master 1, Paris-7) about the development of a user-friendly teaching-oriented documentation for the learn-ocaml platform; Olivier Martinot (Licence 3, Paris 7) about the implementation of a set of efficient incrementalised combinators for list processing in cache-transfer style.

Alexis Saurin supervised Rémi Nollet (M2 LMFI, Paris-7, 2016, with Christine Tasson) on proof theory of temporal logics and functional reactive programming type systems; Ikram Cherigi (M2 LMFI, Paris-7, 2018, with Boban Velickovic) about classical realisability and forcing in set theory; Xavier Onfroy (M2 LMFI, Paris-7, 2018) on formalisation of circular proofs in fixed-point logics and the decidability of validity; Kostia Chardonnet (M1 MPRI, Paris-7, 2018) on call-by-need calculus, degrees of lazyness and probabilistic λ-calculus.

Matthieu Sozeau supervised the M2 internships of Gabriel Lewertowsky (with Nicolas Tabareau, MPRI, 2015) on "Nominal Sets in Coq/SSReflect", Cyprien Mangin (MPRI, 2015) on "Eliminating Dependent Pattern Matching in Coq" and Théo Winterhalter (MPRI, 2017) on "Universes and Reflection".

3.5 General audience actions

Pierre-Louis Curien gave a talk in the Lycée Georges Dumézil (Vernon, Eure, May 2018) on computer bugs and their prevention, on the occasion of the 50th anniversary of this high school.

Hugo Herbelin wrote with Sandrine Blazy and Pierre Castéran in 2018 an introduction to Coq for engineers edited by Techniques de l'Ingénieur [14].

Pierre Letouzey and Yann Régis-Gianas took part in the "Salon du jeu mathématiques" at Saint-Sulpice, Paris, in 2015.

Jean-Jacques Lévy is member of the Inria-Alumni's executive committee (4 meetings in 2018) and organised the session about the transparency of algorithms. He was invited by the French Academy of Sciences to participate to the 2018 Hangzhou International Human Resources Exchanges and Cooperation Conference (Hangzhou, November 2018). He gave a presentation about "Science et Informatique" at the primary school le Coteau, Vaucresson (November 2018). He talked about "L'informatique en 4 temps" at the Alumni-UniThé seminar at Inria Bordeaux (October 2018).

Yann Régis-Gianas co-organised the "Journée Francilienne de Programmation", a programming contest between undergraduate students of three universities of Paris (UPD, UPMC, UPS) in 2015, 2016, 2017 and 2018. He organised the animation of the (computer science part of the) "Fête de la Science" event at the University Paris 7 in 2015 and 2016. He gave several presentations about "What is programming?" in primary and high schools of Paris and its region during the whole period of evaluation. Since 2018, he is the project leader of the Learn-OCaml project whose purpose is to support teaching the OCaml programming language worldwide. In collaboration with Roberto Di Cosmo and Ralf Treinen, he has created a MOOC about the OCaml programming language (three editions, in 2015, 2016 and 2018).

3.6 Visibility

Organisation of scientific events. Pierre-Louis Curien organised a day of homage to the memory of Maurice Nivat on February 6, 2018, at Univ. Paris 7.

Yves Guiraud organised, with Philippe Malbos (Univ. Lyon 1) and Samuel Mimram (École Polytechnique), four editions of the Higher-Dimensional Rewriting and Algebra (HDRA) workshop, in Warwaw (2015), Porto (2016) and Oxford (2017, 2018). He organised, with Dimitri Ara (Univ. Aix-Marseille) and Samuel Mimram (École Polytechnique), the Categories in Homotopy and Rewriting international conference, at the CIRM in Marseille (80 participants, 2017).

Yves Guiraud and Alexis Saurin, with Christine Tasson from IRIF, organised the annual meeting of the Géocal and LAC working groups of the GDR Informatique Mathématique (Paris, 2016).

Pierre Letouzey, Yann Régis-Gianas and Matthieu Sozeau organised the "École de Printemps d'Informatique Théorique 2015" in Fréjus about proof of programs.

Yann Régis-Gianas was multimedia chair of the organising committee of POPL 2017, Paris.

Alexis Saurin organised and co-chaired with David Baelde the Paris workshop in Oxford, UK, collocated with FLoC 2018.

Matthieu Sozeau co-organised and chaired the first Coq for Programming Languages (CoqPL) workshop, collocated with POPL 2015 in Mumbai, India.

Matthieu Sozeau co-organised with Nicolas Tabareau the 3rd Coq Implementors Workshop in Le Croisic, France, in June 2017. It included presentations from developers, both from France and abroad, and a large amount of hacking.

Matthieu Sozeau co-organised and co-chaired with Nicolas Tabareau the Coq Workshop 2018 in Oxford, UK, collocated with FLoC 2018.

Programme committees. Pierre-Louis Curien is member of the programme committee of FSCD 2020.

Yves Guiraud was member of the programme committee of HDRA 2015, 2016, 2017 and 2018, and of IWC 2016.

Hugo Herbelin was member of the programme committees of the conference FSCD 2017, of the TYPES 2017 venue, as well as of the PxTP 2017 and CoqPL 2018 workshops, and of the conference POPL 2019.

Yann Régis-Gianas was member of the programme committee of JFLA 2018, JFLA 2019, and PPDP 2018.

Alexis Saurin was member of the programme committee of the workshop Coinduction in Type Theory (Chambéry, 2017), and of the workshop on Trends in Linear Logic and Applications (Oxford, 2017, satellite event of FSCD).

Matthieu Sozeau was member of the programme committees of FSCD 2016, ITP 2016 and CoqPL 2016, CoqPL 2017, ITP 2018 in Oxford during FLoC 2018, and of the 13th Workshop on Logical and Semantic Frameworks with Applications, Fortaleza, Brazil, 2018.

Steering committees. Pierre-Louis Curien is member of the steering committee of the international workshop Games for Logic and Programming Languages (GaLoP), since 2006.

Pierre-Louis Curien and Hugo Herbelin were members of the steering committee of the conference *Typed Lambda Calculi and Applications* (TLCA) until its merge with the conference *Rewriting Techniques and Applications* in the new conference *Formal Structures for Computation and Deduction* (FSCD), whose first edition was held in Porto in 2016. Hugo Herbelin was a member of the steering committee of FSCD until September 2017.

Hugo Herbelin is a member of the steering committee of the conference TYPES since 2011.

Matthieu Sozeau is member of the steering committee of the Dependently Typed Programming international workshop (DTP) since 2012.

Editorial boards. Pierre-Louis Curien is editor in chief of the Cambridge Univ. Press journal Mathematical Structures in Computer Science (since January 2016).

Hugo Herbelin, Pierre Letouzey and Matthieu Sozeau were co-editors of the post-proceedings of the conference TYPES 2014, published in October 2015 as LIPICS volume 39.

Alexis Saurin is editing a special issue of MSCS dedicated to contributions in honour of Dale Miller for his 60th birthday (almost completed at the date of this report).

Invited talks. Pierre-Louis Curien gave invited talks at the GALOP 2015 (Games in Logic and Programming) Workshop (London, 2015) on "Sequential algorithms: old and new"; at the "Journées Calculabilité 2016" (GDR IM, Nice, 2016), on "Comportement séquentiel et interactif des schémas de récursion primitive"; at the annual meeting of the Géocal and LAC working groups of the GDR Informatique Mathématique (Paris, 2016) on "Opérades cycliques, arbres non enracinés et cohérence"; at the annual meeting of the international ANR project Pace (between Univ. of Bologna, ENS Lyon and Shanghai Jiaotong Univ.) on "Cyclic Operads, Unrooted Trees, and Coherence"' (Shanghai, 2016); at the conference Categories for Homotopy Theory and Rewriting on "A syntactic approach to polynomial functors, polynomial monads and opetopes" (2017). He gave talks on the legacy of Maurice Nivat at two special events organised to honour his memory: special sessions in the Journées du GDR IM (École Polytechnique, 2018), and at ICALP 2018 (Prague, 2018).

Hugo Herbelin gave in 2016 an invited talk on "Proving with side-effects" at the Days in Logic meeting in Lisbon, Portugal. He gave an invited talk on computing with Gôdel's completeness theorem using side effects at the workshop Proof, Computation and Complexity in Bonn, 2018.

Jean-Jacques Lévy gave an invited talk on "Strongly connected components in graphs, Formal proof of Tarjan 1972 algorithm" at the LTP (Langages, Types et Preuves) day, Saclay (2016). He gave a talk at the first Inria-CAS joint project VIP meeting, Paris (November 2018) on "Comparing a formal proof in Why3, Coq and Isabelle".

Yann Régis-Gianas gave an invited talk at the Conference for Trends in Functional Programming in Education on the OCaml MOOC (2017). He gave an invited talk about copatterns in OCaml at the "Logique, Types et Preuves" workshop of the GDR GPL in 2018.

Alexis Saurin gave an invited talk, in the form of a distilled tutorial, at WoC 2015, affiliated to ETAPS 2015 on "Logical by need".

Matthieu Sozeau gave invited talks at the HoTT/UF workshop (Warsaw, 2016) on "Coq support for Homotopy Type Theory"; at the DeepSpec kickoff meeting (Princeton, 2016), on "Coq 8.6" (together with Maxime Dénès); at the International Conference on Mathematical Software (Berlin, 2016), on "Coq for HoTT"; at the Categorical Logic and Univalent Foundations workshop (Leeds, UK, 2016), on "Forcing Translations in Type Theory"; at the Coq Workshop (Nancy, 2016), on "Coq 8.6". He gave an invited talk on "The Predicative, Polymorphic, Cumulative Calculus of Inductive Constructions" at the TYPES 2018 International Conference on Types for Proofs and Programs in Braga, Portugal, 2018. He gave an invited seminar entitled "Programmer en Coq" at the Collège de France, on 2018, part of Xavier Leroy's lectures on the Curry-Howard Isomorphism.

Research stays abroad. Pierre-Louis Curien visited the Category Theory group at Macquarie Univ. for two months in 2016 (collaborative work on the combinatorial structure of type dependency). He visited ECNU (Shanghai) for a month in 2017, and a month from in 2018 (collaborations with Yuxin Deng and Min Zhang) as invited professor. He visited the Institute of Mathematics of the Serbian Academy of Sciences in Belgrade in 2018 for a week (collaboration with Zoran Petrić and other coauthors).

Jean-Jacques Lévy visited the Institute of Software of Chinese Academy of Sciences (ISCAS) in 2017 (project VIP and on-going work with Ran Chen) during two weeks. He gave talks at ISCAS hosted by Ying Jiang, and during a third week at ECNU Shanghai hosted by Min Zhang, USTC Suzhou (University of Science and Technology of China) hosted by Xinyu Feng, Nankai Univ. in Tianjin hosted by Chunfu Jia.

Hugo Herbelin participated to the Types, Sets and Constructions Trimester Program at the Hausdorff Research Institute of Mathematics in Bonn, in 2018.

Scientific expertise. Pierre-Louis Curien has been member of the "Comité de Sélection" for a professor position in mathematics at Univ. Paris 7 in 2015. He has been member of the "Comité de Sélection" for a professor position in discrete mathematics at Univ. Univ. Paul Sabatier in Toulouse in 2016. He has been an expert for a hiring committee for an assistant professor position in Logic, Computation and Programming at Stockholm Univ. (2018).

Pierre-Louis Curien and Yves Guiraud have been members of the "Comité de sélection" for an assistant professor position in mathematical foundations of computer science at Univ. Paris 7 in 2017.

Yves Guiraud is a reviewer for the AMS MathReviews (since 2017). He has been a reviewer for the Université Catholique de Louvain (2018).

Hugo Herbelin has been member of the "Comité de Sélection" for a starting researcher position at INRIA Saclay in 2016. In 2018, he has been a reviewer for FWF (Austrian research funding agency) and NKFI (Hungarian research funding agency).

Hugo Herbelin and Yann Régis-Gianas have been members in 2016 of the "Comité de Sélection" for an assistant professor position at CNAM in Paris.

Yann Régis-Gianas has been member of the "Comité de Sélection" for an assistant professor position at Univ. Paris Sud in 2016. He has been member of the "Comité de Sélection" for one assistant professor position at IRIF in Paris in 2017 and for two assistant professor positions at IRIF in Paris in 2018.

Research administration. Pierre-Louis Curien is a member of the Scientific Council of the CIRM (Centre International de Rencontres Mathématiques) (since 2013).

Pierre-Louis Curien, Yves Guiraud and Hugo Herbelin are members of the scientific council of the Computer Science department of Univ. Paris 7 (since January 2016).

Yves Guiraud is the head of the "Preuves, Programmes et Systèmes" (PPS) pole of the IRIF laboratory (since April 2016), and a member of the IRIF direction council (since September 2017). He was a member of the IRIF laboratory council (January 2016 - December 2017), and the developer and maintainer of the new IRIF website (September 2015 - December 2018).

Hugo Herbelin is the head of the "Preuves et Programmes" thematic team of IRIF (since September 2017).

Since 2018, Yann Régis-Gianas is a member of the Executive Committee of the OCaml Foundation, acting as a representative of the teaching community. Since 2018, in collaboration with Emmanuel Chailloux (UPMC), he is organising the next four years of IRILL, an initiative about innovation in free software.

4 Funding

4.1 National initiatives

Pierre-Louis Curien (coordinator) and Yves Guiraud (local coordinator) have been members of the three-year Focal project of the IDEX Sorbonne Paris Cité (July 2013 to June 2016). This project concerned the interactions between higher rewriting and combinatorial and higher algebra, and was joint with Univ. Paris 13. It funded the PhD grant of Cyrille Chenavier.

Pierre-Louis Curien (coordinator) and Yves Guiraud (local coordinator) have been members of the four-year Cathre ANR project (January 2014 to December 2017). This project investigated the general theory of higher categories, the development of a library of rewriting methods for algebra, and applications in the fields of combinatorial linear algebra, combinatorial group theory and theoretical computer science. This project was joint with Univ. Paris 13, École Polytechnique, Univ. Lyon 1, Univ. Saint-Étienne), Univ. Aix-Marseille and Univ. Toulouse 3. It funded the PhD grant of Maxime Lucas.

Pierre-Louis Curien, Yves Guiraud, Hugo Herbelin and Alexis Saurin are members of the GDR Informatique Mathématique, in the LHC (Logique, homotopie, catégories) and Scalp (Structures formelles pour le calcul et les preuves) working groups. Alexis Saurin is the coordinator of the Scalp working group.

Pierre-Louis Curien, Yves Guiraud (local coordinator) and Matthieu Sozeau are members of the GDR Topologie Algébrique, federating French researchers working on classical topics of algebraic topology and homological algebra, such as homotopy theory, group homology, K-theory, deformation theory, and on more recent interactions of topology with other themes, such as higher categories and theoretical computer science.

Yves Guiraud is member of the GDR Tresses, federating French researchers working on algebraic, algorithmic and topological aspects of braid groups, low-dimensional topology, and connected subjects.

Hugo Herbelin was the coordinator of the PPS site of the ANR project Récré (January 2012 to mid 2016), about realisability and rewriting, with applications to proving with side-effects and concurrency.

Hugo Herbelin, Matthieu Sozeau and Yann Régis-Gianas have been members of the ANR project Paral-ITP (November 2011 to June 2015), aimed at preparing the Coq and Isabelle interactive theorem provers to a new generation of user interfaces thanks to massive parallelism and incremental type-checking.

Since 2014, Yann Régis-Gianas collaborates with Mitsubishi Rennes on the topic of differential semantics; this collaboration led to the CIFRE grant for the PhD of Thibaut Girka. Since 2018, he collaborates with ANSSI on the topic of certified ful programming in Coq. Since 2016, he is a member of the ANR project COLIS dedicated to the verification of Linux distribution installation scripts. This project is joint with Univ. Paris-Sud and Univ. Lille.

Yann Régis-Gianas and Alexis Saurin (coordinator) are members of the four-year ANR project RAPIDO (started in January 2015), investigating the use of proof-theoretical methods to reason and program on infinite data objects. The goal of the project is to develop logical systems capturing infinite proofs (proof systems with least and greatest fixpoints as well as infinitary proof systems), to design and study programming languages for manipulating infinite data, such as streams, both from a syntactical and semantical point of view. This project is joint with ENS Cachan and ENS Lyon.

Matthieu Sozeau is a member of the CoqHoTT ERC project (2015-2020) led by Nicolas Tabareau (Gallinette team, Inria Nantes). The PhD grant of Antoine Allioux, the post-doctoral grant of Eric Finster, and Amin Timany's two-month visit are funded by the CoqHoTT ERC.

4.2 European projects

Hugo Herbelin is a deputy representative of France in the COST action EUTYPES (European research network on types for programming and verification, 2016-2020, led by Herman Geuvers, Univ. Nijmegen). This action aims at promoting: (1) the synergy between theoretical computer scientists, logicians and mathematicians to develop new foundations for type theory, for example as based on the recent development of homotopy type theory; (2) the joint development of type-theoretic tools like proof assistants and integrated programming environments; (3) the study of dependent types for programming and its deployment in software development; (4) the study of dependent types for verification and its deployment in software analysis and verification.

4.3 Industrial contracts

Since 2014, Yann Régis-Gianas collaborates with MERCE (Mitsubishi Electrics Research and Development, located in Rennes) on the topic of the certification of software evolutions. This contract funded a PhD grant.

4.4 Inria Project Labs, Exploratory Research Actions and Technological Development Actions

The Coq development was supported by the ADT Coq (ADT-108) during the evaluation period, which helped recruit an engineer, Matej Košic, in 2015-2016 in the team who then moved to Marelle for another year. He focused on continuous integration and packaging. In 2017 and 2018, the DGD-T gave 15k euros of funding (9k for Marelle, 6k for πr^2) and 24k shared among Gallinette, πr^2 and Marelle. This funding helps with dissemination missions and the organisation of developer and researcher workshops (traditionally attached to POPL and ITP), and working groups, which have since moved to a more decentralised setup through video-conferencing.

4.5 Associated teams and other international projects

Pierre-Louis Curien is a member of the CRECOGI associate team (2015-2018, renewed in 2018), coordinated by Ugo Dal Lago (team FoCUS, Inria Sophia and Bologna) and by Ichiro Hasuo (NII, Tokyo). Presentation of CRECOGI (Concurrent, Resourceful and full Computation, by Geometry of Interaction): Game semantics and geometry of interaction (GoI) are two closely related frameworks whose strength is to have the characters of both a denotational and an operational semantics. This project aims at investigating the application of GoI to concurrent, resourceful, and effectful computation, thus paving the way to the deployment of GoI-based correct-by-construction compilers in real-world software developments in fields like (massively parallel) high-performance computing, embedded and cyberphysical systems, and big data.

Pierre-Louis Curien is principal investigator on the French side for the Inria - Chinese Academy of Sciences project “Verification, Interaction, and Proofs”. The principal investigator on the Chinese side is Ying Jiang, from the Institute of Software in Beijing. The participants of the project on the French side are Pierre-Louis Curien and Jean-Jacques Lévy, as well as other members of IRIF, and Gilles Dowek (Deducteam team). The project funded the postdoc of Kailiang Ji at IRIF (December 2017 to March 2019). Presentation of VIP: The aim of this project is to bring specialists of model-checking and proof-checking together (the “V” and the “P” of the acronym). Applications in the realm of distributed computation, or concurrency theory (the “I”) are particularly targeted.

Pierre-Louis Curien and Tarmo Uustalu (Tallin Univ. of Technology and Univ. of Reykjavik) are the principal investigators of the bilateral Parrott project (Partenariat Hubert Curien) “Mathematical structures for dualities in programming languages” (2017 and 2018).

4.6 Other funding

Pierre-Louis Curien participated to the ANR International French-Chinese project LOCALI (Logical Approach to Novel Computational Paradigms), coordinated by Gilles Dowek (Project-team Deducteam). This project ended in July 2017.

Matthieu Sozeau is part of an international collaboration network CSEC “Certified Software Engineering in Coq” funded by Inria Chile, Conicyt and the CoqHoTT ERC, which officially started in early 2018. The participants include Eric Tanter (primary investigator) and Nicolas Tabareau.

5 Follow up to the previous evaluation

We answer to comments and suggestions from the 2015 evaluation report of the team.

Principal strengths and weaknesses of the project (pp. 35-36).

- *"Despite the stated interest in dependently-typed programming languages, the emphasis of the project seems to be on the proof theory side."*
  There was a strong focus during the evaluation period on certified programming with dependent types across the team. We lifted proof-theory results to the dependent case (lazy dependent choice), we studied programming with effects in dependent type theory (Guillaume Claret's PhD, Yann Régis-Gianas' collaboration with ANSSI), and we enhanced the treatment of dependent pattern-matching and recursion (Boutillier's thesis, the EQUATIONS package) and of how to deal with proof-irrelevance. Finally we collaborate on efficient compilation of dependently-typed programs in the setting of the CertiCoq project.
- *"The project can increase its impact by further strengthening the connections among the subgroups within the team. For example, avenues for cross-fertilisation between higher-dimensional rewriting and homotopy type theory should be investigated."*
  We agree that there are similarities between the two research fields, such as polygraphic resolutions vs. higher inductive types, or weak representations vs. dependent types, but formal dictionaries have proved to be quite hard to build. As a first step to understand how to relate these two worlds, a postdoc was funded for 2017-2019, and a PhD was started in 2018 with the objective to better understand how to formalise basic algebraic structures in homotopy type theory.
- *"In a similar vein, more could be done to transfer advances in the metatheory of Coq to the design and implementation of the proof assistant; a possible example is certified extraction."*
  The MetaCoq project aims to do just that: providing a rigorous, mechanised model of the evolving type theory and linking it to a reference implementation. This is used as a target theory, to extend Coq through verified translations, which is a focus of the CoqHoTT project. It is also the source of an extraction procedure that is already used in the CertiCoq compiler; its correctness proof is ongoing work.
- *"Finally industrial interaction and partnership, which appears sporadic [...], could be developed more systematically".*
  We have solid contacts with Mitsubishi and Tweag I/O. We are open to collaborations which would emerge naturally. We indirectly support industrial use of Coq as part of our collaboration with ANSSI which is in charge of validating industrial applications certified in Coq. Of course, more industrial interactions in the context of Coq can go through a structure like the Coq consortium.

Plan for the next period (pp. 36-37).

- *"A natural and relevant direction concerns algebraic effects in the sense of Plotkin, Power and Hyland, and developed by Melliès, Staton and others. We would suggest that this topic be added to the research plan."*
  We were not familiar enough with algebraic effects to add it explicitly to our research plan. We are focusing on effects where logic and programming meet and it is not clear what algebraic effects could mean in the context of logic. Nevertheless we stay informed about this topic. For instance, Hugo Herbelin has been a reviewer of the PhD of Jirka Maršík on algebraic effects.
- *"A new objective of the plan, Reasoning and programming with infinite data, is in line with the team's general goal of finding logical explications of computational phenomena [...]. We would encourage interactions with research groups elsewhere working on the same theme."*
  Several connections were made: first, the ANR project RAPIDO helped building strong

relations with researchers at LSV and LIP over the whole evaluation period; second, various connections were made, most notably with groups in the UK, Sweden, Denmark, Italy and Japan; last, through the organisation of a workshop in July 2018, additional interactions were created (A. Saurin is involved in a follow up workshop taking place in Sweden late 2019).

Opportunities and risks/difficulties faced by the project (p. 37).

- *"We recommend that the interface between developers and users [of Coq] be improved with the appropriate engineering support."*
  Inria has been of great help improving engineering support: two engineers are now working full-time on Coq development and have growing expertise on the system, interacting with both researchers developing the system and users. The last four years have seen an increase of interactions between users and developers, through public communication channels, continued animation of workshops and the introduction of "coding sprints", an open package index and the coq-community effort. We believe that the interface has greatly improved in that time frame, even if much remains to do.
- *"We sensed the willingness of the relatively small group of researchers working on these topics [(effective higher algebra)] to engage and establish links with the rest of the team; we would strongly encourage such interactions."*
  See above (second item of "Principal strengths and weaknesses of the project").
- *"Finally as the scientific leader approaches retirement, there is an urgent need for succession planning and the grooming of potential leaders."*
  The current scientific leader will retire in October 2019. Considering that the team is ten years old, the members will propose a new scientific project in a close future, in agreement with the direction of the INRIA Paris center.

Recommended actions and suggested measures of success (p. 38).

- *"To address the shortage of qualified engineers, we recommend that INRIA establish longer fixed-term (4 or 5 years) advanced engineer positions linked to software development. During the recruitment process, the applicants should be evaluated as software engineers, not academic researchers."*
  This wish has been granted by Inria, which is now employing two full-time engineers with PhDs in the field (Maxime Dénès in Nice and Vincent Laporte in Rennes), as part of the Coq consortium. The hope is to be able to secure external funding from academic and industrial partners in the future, as the development of Coq benefits to a community far larger than Inria. Also, we note that the new INRIA CEO considers the possibility for a few research teams with heavy development duties to host a full-time research engineer. We are obviously particularly interested by such an opportunity.
- *"Seriously consider moving to a github-like model in which external 'pull requests' are actively encouraged and accepted, particularly in areas outside of the proof-critical kernel. INRIA should also seriously consider anything that could further facilitate community involvement, for example the integration of issue tracking with source control."*
  The entire development has moved to the github platform and external contributions are encouraged and welcomed (to wit, the last versions had about 50 contributors from all around the world). The development team actively uses source control, discussions and reviews of pull-requests, issue tracking and continuous integration. The move of the bugtracker from BugZilla to Github was performed by Théo Zimmermann and resulted in a useful empirical software engineering study.
- *"Encourage design input from programming language and compiler experts, not just members of the* πr^2 *team. Much of the Coq system lives above the level of the kernel type theory. Teams Marelle and Gallium, and external experts, could have a role here."*
  Examples of such design input appear more often today. The collaboration with the CertiCoq

team led by Andrew Appel is such an example, and motivates for example a change of representation of the pattern-matching construct that Hugo Herbelin is implementing in the kernel. The design of the new proof-irrelevance system was done in collaboration with members of the Gallinette team and a member of the Agda development team. In general, the Coq development team is made of engineers and researchers at Inria and elsewhere who collaborate on the design of all components of the tool, some of which πr^2 has little expertise on, such as user interfaces. At the level of the ecosystem, Coq's new package manager is based on OCaml's (thanks to work by Guillaume Claret) and the new coq-community effort launched by Théo Zimmermann both draws from the elm-community initiative and inspires the ocaml-community effort.

6 Objectives for the next four years

The project-team will end soon, as its leader retires in late 2019. The discussion on a new scientific project has started and will be resumed after the writing of this evaluation report. Here, we list the research objectives that we have already identified in the continuity of the scientific programme of πr^2, including some items that emphasise convergences between the present axes of research.

Proof theory and the Curry-Howard correspondence.

- *Effects in logic*. We intend to pursue and disseminate our results on proving with side effects, connecting forcing and memory assignment. We aim at a unifying framework characterising programming effects and their semantics, as well as logical translations and their underlying computational contents, addressing also the case of realisability.
- *Axiom of choice*. We intend to investigate further the computational contents of different variants of the axiom of choice, including the classical extensional axiom of choice, last missing piece not entering the proof-as-program correspondence yet.
- *A computational approach to reverse mathematics*. Reverse mathematics relate theorems of similar logical strength, but, from a proof-as-program point of view, reverse mathematics results can also be seen as transformations of programs, which we shall investigate.

Homotopy type theory.

- *Cubical type theory*. Cubical type theory is an alternative approach to the characterisation of (higher-dimensional) equality in type theory. We intend to investigate its properties and the connections with iterated realisability. As a prospective direction, we plan to analyse the structure of higher-dimensional equality proofs as computational data.
- *Formalisation of algebraic structures*. Around the PhD thesis of Antoine Allioux, the similarities of higher algebra and homotopy type theory will continue to be investigated: in particular, the potential connexions between polygraphic resolutions and higher inductive types, and also between weak higher representations (i.e. pseudofunctors with values into higher categories) and dependent types.

Certified programming.

- *Coq as a state-of-the art dependently-typed programming language*. Coq's functional programming language provides primitives to express a wide variety of programs, however we are in need of higher-level tools to make programming and proving with dependent types more accessible in the system. Building on our expertise on compilation of pattern-matching, co-patterns and (co-)recursion, we plan to provide more robust, flexible and expressive tools for the definition of programs and their proofs. This work goes hand-in-hand with continued foundational advances in the core theory: we will investigate the use of definitional proof-irrelevance to ease programming with refinement types and the introduction of inductive-inductive and inductive-recursive types in the core type theory in particular.

- *Coq as a general purpose programming language*. Effectful programming in Coq is possible [63]: not only can we modularly define programs that interact with their environment, but we know how to reason about these programs inside Coq. The question is now to scale up these techniques to turn Coq into a realistic programming language, following the same path as the Haskell programming language took in the 1990s. To achieve this project, we need to develop a proper compilation chain for Coq programs and a well-suited *Foreign Function Interface* to be able to call system primitives and external libraries. Once this will be done, an important and difficult work will be the design of a standard library (file manipulation, networking, OS interaction) with rich specifications.
- *Gallina as a meta-programming language for Coq*. Today, Coq is made of three languages: Gallina for specifications and programming, Ltac for proof scripting and Vernacular for meta-level definitions. This multiplication of languages increases the heterogeneity of the system interface, it creates unnecessary learning difficulties, and it leads to redundancy in the code base. The Mtac2 project is demonstrating that Gallina is a good replacement to Ltac: it provides a typed language with clear semantics which significantly improves the quality of proof scripts. Besides, the clear specification of Mtac2 allowed us to start the implementation of a JIT compiler for Mtac2 (reusing a machinery for native computation already present in Coq). That project was out of reach if applied to Ltac. Continuing in that direction, the next step is to replace Vernacular by Gallina. This project is ambitious because Vernacular has a large range of mechanisms (small-scale and large-scale definitions management, notations, etc.), but if we can control these mechanisms programmatically from Gallina itself, it would be possible to naturally write generative programs, which are encoded using typeclass instances today.
- *Certifiying Coq*. In another direction, using the deep embedding of Coq in Coq provided by the MetaCoq project, we can hope to (partially) verify the implementation of Coq itself inside Coq and provide a mechanised specification of its theory. On top of that, it will be possible to study and verify extensions of that theory (i.e. through translations from higher-level type theories or DSLs) and the extraction and compilation phases from Coq terms themselves. We plan to complete these projects, providing a fully-certified program development chain from high-level dependently-typed programs down to machine code.
- *A certification framework for differential functional programs*. Even though cache-transfer-style static differentiation provides a way to compose incremental programs in a higher-order setting, the incremental version of base operations (over lists, trees, etc.) still has to be implemented. Manual incrementalisation of programs is error-prone and difficult because (i) if a data type has n constructors and m kinds of change, writing an incremental program requires nm cases; (ii) change application is often undefined (think about removing an element from an empty list) and this leads to many limit cases in incrementalised programs and to complex preconditions; (iii) incrementalised higher-order combinators, typically iterators, can exploit the properties of their input functions to compute their own differential more efficiency; (iv) to be efficient, incremental programs use complex data structures. Coq is a programming language of choice to deal with these problems, because it allows program-proof co-design leading to programs which are correct by construction. We have started the development of a standard library of certified incrementalised primitives named ΔCoq as well as a tool named ΔCaml which allows an OCaml program to automatically integrate these primitives in larger incrementalised programs.

Reasoning and programming with infinite data.

- *Pursuing the investigation of circular proofs*. We aim at developing more expressive validity conditions and more flexible proof systems for infinite and circular proofs (bouncing threads in proof nets and natural deduction systems), and at developing denotational semantics of non-wellfounded proofs of μMALL.

- *Investigating applications of circular proofs to programming*. We wish to undertake a formal study of the relationships between various ways to validate/ensure productivity of (co)inductive computation/validity of (co)inductive reasoning: thread-based in connection with non-wellfounded proofs, size-change principle in connection with program termination, modality based in connection with synchronous programming. We also want to investigate whether circular proofs and thread-based validity can offer interesting alternatives to guard condition for Coq's coinductve types.
- *Applications of circular proofs*. We plan to investigate in the near future the use of circular proofs in modelling the introduction of (co)inductive types in quantum programming. This is a joint project with Benoît Valiron (Centrale Supélec): a previous work by Valiron considers a language of reversible patterns (as a core system for quantum programming), which can be viewed as MALL isos in a language where MALL is extended with an inductive type of lists. Non-overlapping and completeness of patterns is achieved in a system reminiscent of circular focusing proofs, and we plan to extend this language with generic (co)inductive type constructions and relate precisely μMALL focusing with the above mentioned type derivations on patterns.

Effective algebra.

- *Generalised polygraphs*. Notions of polygraphs are now well known for structures such as monoids, categories or associative algebras, which have the common property to be monoids is specific monoidal categories. We plan to extend the definition of polygraph to the whole class of monoids in monoidal categories, also including operads, Lawvere theories, higher-order theories (including λ-calculus and type theory), etc. This will pave the way to new rewriting methods to compute on these objects (collaboration with Marcelo Fiore, Univ. Cambridge).
- *Garside methods in rewriting*. Originating in the study of braid groups, Garside theory provides alternative ways of computing normal forms and resolutions for monoids [DDG+15]. Building on [21], we will continue the integration of Garside methods into rewriting theory, with a view towards an improved completion procedure that can also add new generators to a given presentation (collaboration with Matthieu Picantin, Univ. Paris 7).
- *New applications in effective algebra*. The new, Garside-improved rewriting methods will be applied to obtain better results and new constructions in effective algebra. The main objective is to compute new resolutions for specific monoids, based on Garside families, leading to smaller results than the known rewriting methods (collaboration with Matthieu Picantin, Univ. Paris 7); this would generalise the results of [26] and potentially lead to new progress around the so-called $\mathsf{K}(\pi, 1)$ conjecture, that is deeply linked to the computation of certain precise polygraphic resolutions of Artin groups (planned collaboration with Stéphane Gaussent, Univ. Saint-Étienne, and Philippe Malbos, Univ. Lyon 1). Also, the new completion procedure should explain the normal forms obtained in [ACR18] for quantum circuits on the CNOT and T gates, and possibly provide normal forms for other sets of gates (planned collaboration with Julien Ross, Univ. Dalhousie).

7 Bibliography of the project-team

Doctoral Dissertations and Habilitation Theses

[1] C. CHENAVIER, The lattice of reduction operators: applications to noncommutative Gröbner bases and homological algebra, Theses, Université paris Diderot, December 2016, [hal:tel-01415910].

[2] G. CLARET, Program in Coq, Theses, Université Paris Diderot - Paris 7, September 2018, [hal:tel-01890983].

[3] A. DOUMANE, On the infinitary proof theory of logics with fixed points, Theses, Université Paris 7 - Diderot, June 2017, [hal:tel-01676953].

[4] T. GIRKA, Differential program semantics, Theses, Université Paris Diderot, July 2018, [hal:tel-01890508].

[5] L. D. C. GONZALEZ HUESCA, Incrementality and effect simulation in the simply typed lambda calculus, Theses, Universite Paris Diderot-Paris VII, November 2015, [hal:tel-01248531].

[6] M. LUCAS, Cubical categories for homotopy and rewriting, Theses, Université Paris 7, Sorbonne Paris Cité, December 2017, [hal:tel-01668359].

[7] É. MIQUEY, Classical realizability and side-effects, Theses, Université Sorbonne Paris Cité - Université Paris Diderot (Paris 7) ; Universidad de la República - Montevideo, Uruguay, November 2017, [hal:tel-01653733].

[8] J. OBRADOVIC, Cyclic operads: syntactic, algebraic and categorified aspects, Theses, Université Paris Diderot - Paris 7 - Sorbonne Paris Cité, September 2017, [hal:tel-01676983].

[9] L. PATEY, The reverse mathematics of Ramsey-type theorems, Theses, Université Paris Diderot (Paris 7) Sorbonne Paris Cité, February 2016, PhD thesis, 268 pages, [hal:tel-01888675].

[10] P.-M. PÉDROT, A Materialist Dialectica, Theses, Paris Diderot, September 2015, [hal:tel-01247085].

Articles in International Journals

[11] M. ANEL, G. BIEDERMANN, E. FINSTER, A. JOYAL, Goodwillie's calculus of functors and higher topos theory, *Journal of topology 11*, 4, December 2018, p. 1100–1132, [doi:10.1112/topo.12082], [hal:hal-01939906].

[12] J.-P. BERNARDY, M. BOESPFLUG, R. R. NEWTON, S. PEYTON JONES, A. SPIWACK, Linear Haskell: practical linearity in a higher-order polymorphic language, *Proceedings of the ACM on Programming Languages 2*, POPL, December 2017, p. 1–29, https://arxiv.org/abs/1710.09756, [doi:10.1145/3158093], [hal:hal-01673536].

[13] L. BIENVENU, L. PATEY, P. SHAFER, On the logical strengths of partial solutions to mathematical problems, *Transactions of the London Mathematical Society 4*, 1, December 2017, p. 30 – 71, [doi:10.1112/tlm3.12001], [hal:hal-01888542].

[14] S. BLAZY, P. CASTÉRAN, H. HERBELIN, L'Assistant de Preuve Coq Table des matières, *Techniques de l'Ingenieur*, August 2017, [hal:hal-01645486].

[15] C. CHENAVIER, Confluence Algebras and Acyclicity of the Koszul Complex, *Algebras and Representation Theory*, 2016, [doi:10.1007/s10468-016-9595-6], [hal:hal-01141738].

[16] C. CHENAVIER, Reduction Operators and Completion of Rewriting Systems, *Journal of Symbolic Computation*, 2017, https://arxiv.org/abs/1605.00174, [hal:hal-01325907].

[17] P.-L. CURIEN, S. MIMRAM, Coherent Presentations of Monoidal Categories, *Logical Methods in Computer Science 13*, 3, September 2017, p. 1–38, https://arxiv.org/abs/1705.03553, [doi:10.23638/LMCS-13(3:31)2017], [hal:hal-01662524].

[18] P.-L. CURIEN, J. OBRADOVIC, A formal language for cyclic operads, *Higher Structures*, November 2017, [hal:hal-01278214].

[19] F. DABROWSKI, F. LOULERGUE, T. PINSARD, A formal semantics of nested atomic sections with thread escape, *Computer Languages, Systems and Structures*, 2015, p. 24, [hal:hal-01143199].

[20] D. DE RAUGLAUDRE, Formal Proof of Banach-Tarski Paradox, *Journal of Formalized Reasoning 10*, 1, October 2017, p. 37–49, [doi:10.6092/issn.1972-5787/6927], [hal:hal-01673378].

[21] P. DEHORNOY, Y. GUIRAUD, Quadratic normalisation in monoids, *International Journal of Algebra and Computation 26*, 5, August 2016, p. 935–972, https://arxiv.org/abs/1504.02717, [doi:10.1142/S0218196716500399], [hal:hal-01141226].

[22] D. D. DZHAFAROV, L. PATEY, R. D. SOLOMON, L. B. WESTRICK, Ramsey's theorem for singletons and strong computable reducibility, *Proceedings of the American Mathematical Society 145*, 3, March 2017, p. 1343 – 1355, [doi:10.1090/proc/13315], [hal:hal-01888750].

[23] D. D. DZHAFAROV, L. PATEY, Coloring trees in reverse mathematics, *Advances in Mathematics 318*, October 2017, p. 497 – 514, [doi:10.1016/j.aim.2017.08.009], [hal:hal-01888777].

[24] F. FAGES, T. MARTINEZ, D. A. ROSENBLUETH, S. SOLIMAN, Influence Networks compared with Reaction Networks: Semantics, Expressivity and Attractors, *IEEE/ACM Transactions on Computational Biology and Bioinformatics PP*, 99, 2018, p. 1–14, [doi:10.1109/TCBB.2018.2805686], [hal:hal-01510216].

[25] E. FRITTAION, L. PATEY, Coloring the rationals in reverse mathematics, *Computability 6*, 4, October 2017, p. 319 – 331, [doi:10.3233/COM-160067], [hal:hal-01888756].

[26] S. GAUSSENT, Y. GUIRAUD, P. MALBOS, Coherent presentations of Artin monoids, *Compositio Mathematica 151*, 5, 2015, p. 957–998, [doi:10.1112/S0010437X14007842], [hal:hal-00682233].

[27] G. GILBERT, J. COCKX, M. SOZEAU, N. TABAREAU, Definitional Proof-Irrelevance without K, *Proceedings of the ACM on Programming Languages*, January 2019, p. 1–28, [doi:10.1145/329031610.1145/3290316], [hal:hal-01859964].

[28] M. GUILLERMO, É. MIQUEY, Classical realizability and arithmetical formulæ, *Mathematical Structures in Computer Science*, 2016, [doi:10.1017/S0960129515000559], [hal:hal-01247989].

[29] Y. GUIRAUD, E. HOFFBECK, P. MALBOS, Convergent presentations and polygraphic resolutions of associative algebras, *Mathematische Zeitschrift*, 2018, 68 pages, [hal:hal-01006220].

[30] Y. GUIRAUD, P. MALBOS, Polygraphs of finite derivation type, *Mathematical Structures in Computer Science 28*, 2, 2018, p. 155–201, https://arxiv.org/abs/1402.2587, [doi:10.1017/S0960129516000220], [hal:hal-00932845].

[31] H. HERBELIN, A dependently-typed construction of semi-simplicial types, *Mathematical Structures in Computer Science 25*, Special issue 05, June 2015, p. 16, [doi:10.1017/S0960129514000528], [hal:hal-00935446].

[32] Y. JIANG, J. LIU, G. DOWEK, K. JI, Towards Combining Model Checking and Proof Checking, *The Computer Journal*, 2019, [hal:hal-01970274].

[33] J.-O. KAISER, B. ZILIANI, R. KREBBERS, Y. RÉGIS-GIANAS, D. DREYER, Mtac2: typed tactics for backward reasoning in Coq, *Proceedings of the ACM on Programming Languages 2*, ICFP, July 2018, p. 1 – 31, [doi:10.1145/3236773], [hal:hal-01890511].

[34] F. LOULERGUE, W. BOUSDIRA, J. TESSON, Calculating Parallel Programs in Coq using List Homomorphisms, *International Journal of Parallel Programming 45*, 2, 2017, p. 300–319, [doi:10.1007/s10766-016-0415-8], [hal:hal-01159182].

[35] M. LUCAS, A coherence theorem for pseudonatural transformations, *Journal of Pure and Applied Algebra 221*, 5, 2017, p. 1146–1217, https://arxiv.org/abs/1508.07807, [doi:10.1016/j.jpaa.2016.09.005], [hal:hal-01191867].

[36] J. OBRADOVIC, P.-L. CURIEN, J. IVANOVIC, Syntactic aspects of hypergraph polytopes, *Journal of Homotopy and Related Structures*, 2017, [hal:hal-01669490].

[37] J. OBRADOVIC, Monoid-like definitions of cyclic operad, *Theory and Applications of Categories*, March 2017, [hal:hal-01987321].

[38] L. PATEY, K. YOKOYAMA, The proof-theoretic strength of Ramsey's theorem for pairs and two colors, *Advances in Mathematics 330*, May 2018, p. 1034 – 1070, [doi:10.1016/j.aim.2018.03.035], [hal:hal-01888655].

[39] L. PATEY, Degrees bounding principles and universal instances in reverse mathematics, *Annals of Pure and Applied Logic 166*, 11, November 2015, p. 1165 – 1185, [doi:10.1016/j.apal.2015.07.003], [hal:hal-01888599].

[40] L. PATEY, Ramsey-type graph coloring and diagonal non-computability, *Archive for Mathematical Logic 54*, 7-8, November 2015, p. 899 – 914, [doi:10.1007/s00153-015-0448-5], [hal:hal-01888596].

[41] L. PATEY, The complexity of satisfaction problems in reverse mathematics, *Lecture Notes in Computer Science 4*, 1, 2015, p. 69–84, [hal:hal-01888582].

[42] L. PATEY, Controlling iterated jumps of solutions to combinatorial problems, *Computability 6*, 1, December 2016, p. 47 – 78, [doi:10.3233/COM-160056], [hal:hal-01888648].

[43] L. PATEY, Open questions about Ramsey-type statements in reverse mathematics, *Bulletin of Symbolic Logic 22*, 02, June 2016, p. 151 – 169, [doi:10.1017/bsl.2015.40], [hal:hal-01888620].

[44] L. PATEY, The strength of the tree theorem for pairs in reverse mathematics, *The Journal of Symbolic Logic 81*, 04, December 2016, p. 1481 – 1499, [doi:10.1017/jsl.2015.80], [hal:hal-01888614].

[45] L. PATEY, The weakness of being cohesive, thin or free in reverse mathematics, *Israël Journal of Mathematics 216*, 2, October 2016, p. 905 – 955, [doi:10.1007/s11856-016-1433-3], [hal:hal-01888606].

[46] L. PATEY, Iterative forcing and hyperimmunity in reverse mathematics, *Computability 6*, 3, August 2017, p. 209 – 221, [doi:10.3233/COM-160062], [hal:hal-01888601].

[47] N. TABAREAU, É. TANTER, M. SOZEAU, Equivalences for Free: Univalent Parametricity for Effective Transport, *Proceedings of the ACM on Programming Languages*, September 2018, p. 1–29, [doi:10.1145/3234615], [hal:hal-01559073].

[48] B. ZILIANI, M. SOZEAU, A comprehensible guide to a new unifier for CIC including universe polymorphism and overloading, *Journal of Functional Programming 27*, 2017, [doi:10.1017/S0956796817000028], [hal:hal-01671925].

International Conferences with Proceedings

[49] A. ANAND, S. BOULIER, C. COHEN, M. SOZEAU, N. TABAREAU, Towards Certified Meta-Programming with Typed Template-Coq, *in : ITP 2018 - 9th Conference on Interactive Theorem Proving*, *LNCS*, *10895*, Springer, p. 20–39, Oxford, United Kingdom, July 2018, [doi:10.1007/978-3-319-94821-8_2], [hal:hal-01809681].

[50] D. BAELDE, A. DOUMANE, A. SAURIN, Least and Greatest Fixed Points in Ludics , *in : 24th EACSL Annual Conference on Computer Science Logic (CSL 2015)*, *24th EACSL Annual Conference on Computer Science Logic (CSL 2015)*, *41*, Leibniz International Proceedings in Informatics (LIPIcs), p. 549–566, Berlin, Germany, September 2015, [doi:10.4230/LIPIcs.CSL.2015.549], [hal:hal-01256773].

[51] M. BAGNOL, A. DOUMANE, A. SAURIN, On the dependencies of logical rules, *in : FOSSACS, 18th International Conference on Foundations of Software Science and Computation Structures*, London, United Kingdom, April 2015, [hal:hal-01110340].

[52] A. BAUER, G. JASON, P. LUMSDAINE, M. SHULMAN, M. SOZEAU, B. SPITTERS, The HoTT Library: A Formalization of Homotopy Type Theory in Coq, *in : CPP'17*, *CPP'17*, ACM, ACM, p. 9, Paris, France, January 2017, [doi:10.1145/3018610.3018615], [hal:hal-01421212].

[53] A. BLANCHARD, N. KOSMATOV, M. LEMERRE, F. LOULERGUE, A case study on formal verification of the anaxagoros hypervisor paging system with frama-C, *in : FMICS 2015 - Formal Methods for Industrial Critical Systems*, N. M. Gudemann M. (editor), *Lecture Notes in Computer Science - LNCS*, *9128*, Springer Verlag, p. 15–30, Oslo, Norway, June 2015. Conference of 20th International Workshop on Formal Methods for Industrial Critical Systems, FMICS 2015 ; Conference Date: 22 June 2015 Through 23 June 2015; Conference Code:118829, [doi:10.1007/978-3-319-19458-5_2], [hal:cea-01834977].

[54] R. CHEN, J.-J. LÉVY, A Semi-automatic Proof of Strong connectivity, *in : 9th Working Conference on Verified Software: Theories, Tools and Experiments (VSTTE)*, Heidelberg, Germany, July 2017, [hal:hal-01632947].

[55] P.-L. CURIEN, M. FIORE, G. MUNCH-MACCAGNONI, A theory of effects and resources: adjunction models and polarised calculi, *in : Principles of Programmming Languages*, *Proceedings POPL 2016*, Saint-Petersbourg, Florida, United States, January 2016, [doi:10.1145/2837614.2837652], [hal:hal-01256092].

[56] F. DABROWSKI, F. LOULERGUE, T. PINSARD, Nested atomic sections with thread escape: Compilation to threads and locks, *in : ACM Symposium on Applied Computing (SAC)*, ACM, Salamanca, Spain, April 2015, [hal:hal-01105093].

[57] A. DOUMANE, Constructive completeness for the linear-time μ-calculus, *in : Conference on Logic in Computer Science 2017*, Reykjavik, Iceland, June 2017, [hal:hal-01430737].

[58] T. GIRKA, D. MENTRÉ, Y. RÉGIS-GIANAS, A Mechanically Checked Generation of Correlating Programs Directed by Structured Syntactic Differences, *in : Automated Technology for Verification and Analysis*, *Lecture Notes in Computer Science*, *9364*, Shanghai, China, October 2015, [doi:10.1007/978-3-319-24953-7_6], [hal:hal-01238704].

[59] T. GIRKA, D. MENTRÉ, Y. RÉGIS-GIANAS, Verifiable Semantic Difference Languages, *in : International Symposium on Principles and Practice of Declarative Programming*, Namur, Belgium, October 2017, [doi:10.1145/3131851.3131870], [hal:hal-01653283].

[60] A. GUESMI, P. CLEMENTE, F. LOULERGUE, P. BERTHOMÉ, Cloud Resources Placement Based on Functional and Non-Functional Requirements, *in : SECRYPT*, SCITEPRESS, Colmar, France, July 2015, [hal:hal-01158134].

[61] G. JABER, G. LEWERTOWSKI, P.-M. PÉDROT, M. SOZEAU, N. TABAREAU, The Definitional Side of the Forcing, *in : Logics in Computer Science*, New York, United States, May 2016, [doi:10.1145/http://dx.doi.org/10.1145/2933575.2935320], [hal:hal-01319066].

[62] P. LAFORGUE, Y. RÉGIS-GIANAS, Copattern matching and first-class observations in OCaml, with a macro, *in : International Symposium on Principles and Practice of Declarative Programming*, Namur, Belgium, October 2017, [doi:10.1145/3131851.3131869], [hal:hal-01653261].

[63] T. LETAN, Y. RÉGIS-GIANAS, P. CHIFFLIER, G. HIET, Modular Verification of Programs with Effects and Effect Handlers in Coq, *in : FM 2018 - 22nd International Symposium on Formal Methods*, *LNCS*, *10951*, Springer, p. 338–354, Oxford, United Kingdom, July 2018, [doi:10.1007/978-3-319-95582-7_20], [hal:hal-01799712].

[64] C. MANGIN, M. SOZEAU, Equations for Hereditary Substitution in Leivant's Predicative System F: A Case Study, *in : Tenth International Workshop on Logical Frameworks and Meta Languages: Theory and Practice*, *EPTCS*, *185*, Berlin, Germany, August 2015, [doi:10.4204/EPTCS.185.5], [hal:hal-01248807].

[65] É. MIQUEY, H. HERBELIN, Realizability Interpretation and Normalization of Typed Call-by-Need λ-calculus With Control, *in : FOSSACS 18 - 21st International Conference on Foundations of Software Science and Computation Structures*, C. Baie, U. D. Lago (editors), *LNCS*, *10803*, Springer, p. 276–292, Thessalonique, Greece, April 2018. https://arxiv.org/abs/1803.00914, [doi:10.1007/978-3-319-89366-2_15], [hal:hal-01624839].

[66] É. MIQUEY, A Classical Sequent Calculus with Dependent Types , *in : 26th European Symposium on Programming*, Uppsala, Sweden, April 2017, [hal:hal-01375977].

[67] P.-M. PÉDROT, A. SAURIN, Classical by-need, *in : European Symposium on Programming*, *European Symposium on Programming*, Eindhoven, Netherlands, April 2016, [hal:hal-01257348].

[68] Y. RÉGIS-GIANAS, N. JEANNEROD, R. TREINEN, Morbig: A Static Parser for POSIX Shell, *in : SLE 2018 - ACM SIGPLAN International Conference on Software Language Engineering*, Boston, United States, November 2018, [doi:10.1145/3276604.3276615], [hal:hal-01890044].

[69] E. RIVAS, Relating Idioms, Arrows and Monads from Monoidal Adjunctions, *in : Seventh Workshop on Mathematically Structured Functional Programming- EPTCS*, *275*, p. 18–33, Oxford, United Kingdom, July 2018, [hal:hal-01946996].

[70] A. TIMANY, M. SOZEAU, Cumulative Inductive Types in Coq, *in : FSCD 2018 - 3rd International Conference on Formal Structures for Computation and Deduction*, Oxford, United Kingdom, July 2018, [doi:10.4230/LIPIcs.FSCD.2018.29], [hal:hal-01952037].

[71] T. WINTERHALTER, M. SOZEAU, N. TABAREAU, Eliminating Reflection from Type Theory: To the Legacy of Martin Hofmann, *in : CPP 2019 - The 8th ACM SIGPLAN International Conference on Certified Programs and Proofs*, ACM, p. 91–103, Lisbonne, Portugal, January 2019, [doi:10.1145/3293880.3294095], [hal:hal-01849166].

[72] B. ZILIANI, M. SOZEAU, A unification algorithm for Coq featuring universe polymorphism and overloading, *in : 20th ACM SIGPLAN International Conference on Functional Programming*, Vancouver, Canada, September 2015, [doi:10.1145/2784731.2784751], [hal:hal-01248804].

International Conferences without Proceedings

[73] A. ANAND, S. BOULIER, N. TABAREAU, M. SOZEAU, Typed Template Coq – Certified Meta-Programming in Coq, *in : CoqPL 2018 - The Fourth International Workshop on Coq for Programming Languages*, p. 1–2, Los Angeles, CA, United States, January 2018, [hal:hal-01671948].

[74] G. CLARET, Y. RÉGIS-GIANAS, Mechanical Verification of Interactive Programs Specified by Use Cases, *in : 3rd IEEE/ACM FME Workshop on Formal Methods in Software Engineering*, Florence, Italy, May 2015, [doi:10.1109/FormaliSE.2015.17], [hal:hal-01255107].

[75] H. HERBELIN, É. MIQUEY, Toward dependent choice: a classical sequent calculus with dependent types, *in : TYPES 2015*, Tallinn, Estonia, May 2015, [hal:hal-01247998].

[76] H. HERBELIN, É. MIQUEY, A continuation-passing-style interpretation of simply-typed call-by-need λ-calculus with control within System F, *in : CL&C'16. Sixth International Workshop on. Classical Logic and Computation*, Porto, Portugal, June 2016, [hal:hal-01302696].

[77] É. MIQUEY, Realizability games for arithmetical formulæ , *in : Worskshop GaLoP2015*, Londres, United Kingdom, April 2015, [hal:hal-01248093].

[78] R. NOLLET, A. SAURIN, C. TASSON, Local validity for circular proofs in linear logic with fixed points: extended version, *in : Computer Science Logic*, Birmingham, United Kingdom, September 2018, [hal:hal-01825477].

[79] T. ZIMMERMANN, H. HERBELIN, Automatic and Transparent Transfer of Theorems along Isomorphisms in the Coq Proof Assistant, *in : Conference on Intelligent Computer Mathematics*, Washington, D.C., United States, 2015. https://arxiv.org/abs/1505.05028, [hal:hal-01152588].

[80] T. ZIMMERMANN, H. HERBELIN, Coq's Prolog and application to defining semi-automatic tactics, *in : Type Theory Based Tools*, Paris, France, January 2017, [hal:hal-01671994].

[81] T. ZIMMERMANN, Challenges in the collaborative development of a complex mathematical software and its ecosystem, *in : OpenSym 2018 - 14th International Symposium on Open Collaboration, 2018*, Paris, France, August 2018, [doi:10.1145/3233391.3233966], [hal:hal-01951322].

National Conferences

[82] R. CHEN, J.-J. LÉVY, Une preuve formelle de l'algorithme de Tarjan-1972 pour trouver les composantes fortement connexes dans un graphe, *in : JFLA 2017 - Vingt-huitièmes Journées Francophones des Langages Applicatifs, Vingt-huitièmes Journées Francophones des Langages Applicatifs*, Gourette, France, January 2017, [hal:hal-01422215].

[83] P. LAFORGUE, Y. RÉGIS-GIANAS, OCaml étendu avec du filtrage par comotifs, *in : JFLA 2018 - Journées Francophones des Langages Applicatifs*, Banyuls sur mer, France, January 2018, [hal:hal-01897456].

[84] F. LOULERGUE, Construction de programmes parallèles en Coq avec des homomorphismes de listes, *in : 14èmes journées Approches Formelles dans l'Assistance au Développement Logiciel (AFADL)*, Bordeaux, France, 2015, [hal:hal-01158137].

[85] F. LOULERGUE, Modèles fonctionnels de MapReduce en Coq, *in : Journées Nationales du GdR GPL*, Bordeaux, France, 2015, [hal:hal-01158138].

Scientific Books or Book Chapters)

[86] A. BOUAJJANI, C. ENEA, K. JI, S. QADEER, On the Completeness of Verifying Message Passing Programs Under Bounded Asynchrony, *in : International Conference on Computer Aided Verification, CAV 2018: Computer Aided Verification*, Springer International Publishing, July 2018, p. 372–391, [hal:hal-01947855].

[87] F. LOULERGUE, W. BOUSDIRA, J. TESSON, Calcul de programmes parallèles avec Coq, *in : Informatique Mathématique*, *collection Alpha*, CNRS Éditions, March 2015, [hal:hal-01107296].

Research Reports

[88] T. GIRKA, D. MENTRÉ, Y. RÉGIS-GIANAS, Oracle-based Differential Operational Semantics (long version), *Research report*, Université Paris Diderot / Sorbonne Paris Cité, October 2016, [hal:hal-01419860].

[89] P. LETOUZEY, Hofstadter's problem for curious readers, *Research report*, Université Paris Diderot ; INRIA Paris-Rocquencourt, September 2015, https://arxiv.org/abs/1509.02479, [hal:hal-01195587].

[90] A. TIMANY, M. SOZEAU, Consistency of the Predicative Calculus of Cumulative Inductive Constructions (pCuIC), *Research Report number RR-9105*, KU Leuven, Belgium ; Inria Paris, October 2017, Version 2 fixes some typos from version 1., [hal:hal-01615123].

Miscellaneous

[91] D. BAELDE, A. DOUMANE, A. SAURIN, Least and Greatest Fixed Points in Ludics, This document corresponds to the long version of a paper accepted for publication at CSL 2015, July 2015, [hal:hal-01178396].

[92] D. BAELDE, A. DOUMANE, A. SAURIN, Infinitary proof theory : the multiplicative additive case , working paper or preprint, June 2016, [hal:hal-01339037].

[93] R. CHEN, C. COHEN, J.-J. LEVY, S. MERZ, L. THERY, Formal Proofs of Tarjan's Algorithm in Why3, Coq, and Isabelle, https://arxiv.org/abs/1810.11979 - working paper or preprint, October 2018, [hal:hal-01906155].

[94] R. CHEN, J.-J. LEVY, Readable semi-automatic formal proofs of Depth-First Search in graphs using Why3, working paper or preprint, November 2015, [hal:hal-01253136].

[95] R. CHEN, J.-J. LEVY, Formal proofs of two algorithms for strongly connected components in graphs, working paper or preprint, November 2016, [hal:hal-01422216].

[96] C. CHENAVIER, A Lattice Formulation of the F 4 Completion Procedure, working paper or preprint, January 2018, [hal:hal-01489200].

[97] C. CHENAVIER, Syzygies among reduction operators, https://arxiv.org/abs/1708.08709 - working paper or preprint, April 2018, [hal:hal-01578555].

[98] T. COQ DEVELOPMENT TEAM, The Coq Proof Assistant, version 8.7.1, December 2017, Official Release, [doi:10.5281/zenodo.1133970], [hal:hal-01673716].

[99] T. COQ DEVELOPMENT TEAM, The Coq Proof Assistant, version 8.8.0, April 2018, Software, [doi:10.5281/zenodo.1219885], [hal:hal-01954564].

[100] P.-L. CURIEN, J. OBRADOVIC, On the various definitions of cyclic operads, working paper or preprint, January 2016, [hal:hal-01254649].

[101] P.-L. CURIEN, J. OBRADOVIC, Categorified cyclic operads, https://arxiv.org/abs/1706.06788 - working paper or preprint, January 2018, [hal:hal-01679682].

[102] A. DOUMANE, D. BAELDE, L. HIRSCHI, A. SAURIN, Towards Completeness via Proof Search in the Linear Time mu-Calculus: The case of Büchi inclusions, working paper or preprint, January 2016, [hal:hal-01275289].

[103] T. GIRKA, Y. RÉGIS-GIANAS, Correlating Oracles, https://archive.softwareheritage.org/swh:1:rev:cccf789c12617208fe188ad3dbc2746d4c884ab7, July 2018, Software, [hal:hal-01831369].

[104] T. GIRKA, correlating_program, https://archive.softwareheritage.org/swh:1:rev:c8fca417ee9eefe25683042192da67470147be07, July 2018, Software, [hal:hal-01831364].

[105] H. HERBELIN, É. MIQUEY, Normalization and continuation-passing-style interpretation of simply-typed call-by-need λ-calculus with control, working paper or preprint, July 2017, [hal:hal-01570987].

[106] H. HERBELIN, É. MIQUEY, Continuation-and-environment-passing style translations: a focus on call-by-need, working paper or preprint, January 2019, [hal:hal-01972846].

[107] C. HO THANH, The equivalence between many-to-one polygraphs and opetopic sets, https://arxiv.org/abs/1806.08645 - working paper or preprint, July 2018, [hal:hal-01946918].

[108] N. JEANNEROD, Y. RÉGIS-GIANAS, R. TREINEN, Having Fun With 31.521 Shell Scripts, working paper or preprint, April 2017, [hal:hal-01513750].

[109] J.-J. LEVY, R. CHEN, Strongly Connected Components in graphs, formal proof of Tarjan1972 algorithm, Groupe de travail LTP du GDR GPL , November 2016, Travail présenté à JFLA 2017, [hal:hal-01422227].

[110] M. LUCAS, A cubical Squier's theorem, working paper or preprint, December 2017, [hal:hal-01662132].

[111] M. LUCAS, Cubical (ω,p)-categories, working paper or preprint, December 2017, [hal:hal-01662127].

[112] C. MANGIN, M. SOZEAU, Equations reloaded, working paper or preprint, July 2018, [hal:hal-01671777].

[113] C. MANGIN, Eliminating Dependent Pattern-Matching in Coq, Mémoire, Université Paris 7 Denis Diderot, August 2015, [hal:hal-01250855].

[114] G. MANZONETTO, A. POLONSKY, A. SAURIN, J. G. SIMONSEN, The Fixed Point Property and a Technique to Harness Double Fixed Point Combinators , working paper or preprint, December 2017, [hal:hal-01672846].

[115] J. OBRADOVIC, The Bénabou-Roubaud monadic descent theorem via string diagrams, working paper or preprint, January 2016, [hal:hal-01254637].

[116] Y. REGIS-GIANAS, N. JEANNEROD, R. TREINEN, Morbig, https://archive.softwareheritage.org/swh:1:dir:eb7770ebd7341d08d8dc17f5257ed17852a015a8, October 2018, Software, [hal:hal-01897572].

[117] T. ZIMMERMANN, A. CASANUEVA ARTÍS, Impact of switching bug trackers: a case study, working paper or preprint, December 2018, [hal:hal-01951176].

8 Other citations

Additional references

[AABS02] Fahd Ali Al-Agl, Ronald Brown, and Richard Steiner. Multiple categories: the equivalence of a globular and a cubical approach. *Adv. Math.*, 170(1):71–118, 2002.

[ACR18] Matthew Amy, Jianxin Chen, and Neil J. Ross. A finite presentation of CNOT-dihedral operators. In *Proceedings 14th International Conference on Quantum Physics and Logic*, volume 266 of *Electron. Proc. Theor. Comput. Sci. (EPTCS)*, pages 84–97. EPTCS, [place of publication not identified], 2018.

[AM11] D. Ara and F. Métayer. The Brown-Golasiński model structure on strict ∞-groupoids revisited. *Homology, Homotopy and Applications*, 13(1):121–142, 2011.

[Ani86] David J. Anick. On the homology of associative algebras. *Trans. Amer. Math. Soc.*, 296(2):641–659, 1986.

[Ber98] Roland Berger. Confluence and Koszulity. *J. Algebra*, 201(1):243–283, 1998.

[BG09] G. Bonfante and Y. Guiraud. Polygraphic programs and polynomial-time functions. *Logical Methods in Computer Science*, 5(2):1–37, 2009.

[BH81a] Ronald Brown and Philip J. Higgins. The equivalence of ∞-groupoids and crossed complexes. *Cahiers Topologie Géom. Différentielle*, 22(4):371–386, 1981.

[BH81b] Ronald Brown and Philip J. Higgins. The equivalence of ω-groupoids and cubical T-complexes. *Cahiers Topologie Géom. Différentielle*, 22(4):349–370, 1981.

[BPT17] Simon Boulier, Pierre-Marie Pédrot, and Nicolas Tabareau. The next 700 syntactical models of type theory. In *Certified Programs and Proofs (CPP 2017)*, pages 182 – 194, Paris, France, January 2017.

[Buc65] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal (An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal).* PhD thesis, Mathematical Institute, University of Innsbruck, Austria, 1965.

[Bur93] A. Burroni. Higher-dimensional word problems with applications to equational logic. *Theoretical Computer Science*, 115(1):43–62, jul 1993.

[CDM13] Cyril Cohen, Maxime Dénès, and Anders Mörtberg. Refinements for Free! In *Certified Programs and Proofs*, pages 147 – 162, Melbourne, Australia, December 2013.

[CFC58] Haskell B. Curry, Robert Feys, and William Craig. *Combinatory Logic*, volume 1. North-Holland, 1958. §9E.

[CH85] Thierry Coquand and G. Huet. Constructions : A Higher Order Proof System for Mechanizing Mathematics. In *EUROCAL'85*, volume 203 of *Lecture Notes in Computer Science*, Linz, 1985. Springer Verlag.

[CH00] Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming (ICFP '00)*, SIGPLAN Notices 35(9), pages 233–243, Montreal, Canada, September 18-21 2000. ACM.

[Coq85] Thierry Coquand. *Une théorie des Constructions*. Dissertation, University Paris 7, January 1985.

[CPM90] Thierry Coquand and C. Paulin-Mohring. Inductively defined types. In Per Martin-Löf and G. Mints, editors, *Proceedings of Colog'88*, volume 417 of *Lecture Notes in Computer Science*. Springer Verlag, 1990.

[Cur34] Haskell B. Curry. Functionality in combinatory logic. *Proceedings of the National Academy of Sciences of the United States of America*, 20:584–590, 1934.

[dB68] Nicolaas de Bruijn. Automath, a language for mathematics. Technical Report 66-WSK-05, Technological University Eindhoven, November 1968.

[DDG+15] Patrick Dehornoy, François Digne, Eddy Godelle, Daan Krammer, and Jean Michel. *Foundations of Garside theory*, volume 22 of *EMS Tracts Math.* European Mathematical Society, 2015.

[Del97] Pierre Deligne. Action du groupe des tresses sur une catégorie. *Invent. Math.*, 128(1):159–175, 1997.

[DLGBL17] Ugo Dal Lago, Francesco Gavazzo, and Paul Blain Levy. Effectful Applicative Bisimilarity: Monads, Relators, and Howe's Method. In *LICS 2017 - ACM/IEEE Symposium on Logic in Computer Science*, pages 1–12, Reykjavik, Iceland, June 2017. IEEE.

[Fau99] Jean-Charles Faugére. A new efficient algorithm for computing Gröbner bases (F_4). *J. Pure Appl. Algebra*, 139(1-3):61–88, 1999. Effective methods in algebraic geometry (Saint-Malo, 1998).

[Gö58] Kurt Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12(3):280–287, December 1958.

[Gen35] Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210,405–431, 1935.

[GM09] Yves Guiraud and Philippe Malbos. Higher-dimensional categories with finite derivation type. *Theory Appl. Categ.*, 22(18):420–478, 2009.

[GM11] Yves Guiraud and Philippe Malbos. Identities among relations for higher-dimensional rewriting systems. *Sémin. Congr.*, 26:145–161, 2011.

[GM12a] Yves Guiraud and Philippe Malbos. Coherence in monoidal track categories. *Math. Structures Comput. Sci.*, 22(6):931–969, 2012.

[GM12b] Yves Guiraud and Philippe Malbos. Higher-dimensional normalisation strategies for acyclicity. *Advances in Mathematics*, 231(3-4):2294–2351, 2012.

[GMM13] Yves Guiraud, Philippe Malbos, and Samuel Mimram. A Homotopical Completion Procedure with Applications to Coherence of Monoids. In Femke Van Raamsdonk, editor, *RTA - 24th International Conference on Rewriting Techniques and Applications - 2013*, volume 21 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 223–238, Eindhoven, Netherlands, June 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Gri90] Timothy G. Griffin. The formulae-as-types notion of control. In *Conf. Record 17th Annual ACM Symp. on Principles of Programming Languages, POPL '90*, pages 47–57, San Francisco, CA, USA, 17-19 Jan 1990, 1990. ACM Press.

[Gui04] Y. Guiraud. *Présentations d'opérades et systèmes de réécriture*. PhD thesis, Univ. Montpellier 2, 2004.

[Gui06a] Y. Guiraud. Termination orders for 3-dimensional rewriting. *Journal of Pure and Applied Algebra*, 207(2):341–371, 2006.

[Gui06b] Y. Guiraud. The three dimensions of proofs. *Annals of Pure and Applied Logic*, 141(1–2):266–295, 2006.

[Gui06c] Y. Guiraud. Two polygraphic presentations of Petri nets. *Theoretical Computer Science*, 360(1–3):124–146, 2006.

[Gui19] Yves Guiraud. *Rewriting methods in homotopical and higher-dimensional algebra*. Habilitation à diriger des recherches, Univ. Paris 7, 2019.

[Her95] Hugo Herbelin. A lambda-calculus structure isomorphic to Gentzen-style sequent calculus structure. In Leszek Pacholski and Jerzy Tiuryn, editors, *Computer Science Logic, 8th International Workshop, CSL '94, Kazimierz, Poland, September 25-30, 1994, Selected Papers*, volume 933 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 1995.

[Her05] Hugo Herbelin. On the degeneracy of sigma-types in presence of computational classical logic. In Pawel Urzyczyn, editor, *Proceedings of TLCA 2005*, volume 3461 of *Lecture Notes in Computer Science*, pages 209–220. Springer, 2005.

[Her12] Hugo Herbelin. A constructive proof of dependent choice, compatible with classical logic. In *Proceedings of the 27th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2012, 25-28 June 2012, Dubrovnik, Croatia*, pages 365–374. IEEE Computer Society, 2012.

[How80] William A. Howard. The formulae-as-types notion of constructions. In *to H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. Academic Press, 1980. Unpublished manuscript of 1969.

[HS13] Hugo Herbelin and Arnaud Spiwack. The rooster and the syntactic bracket. In Ralph Matthes and Aleksy Schubert, editors, *19th International Conference on Types for Proofs and Programs, TYPES 2013, April 22-26, 2013, Toulouse, France*, volume 26 of *LIPIcs*, pages 169–187. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[Kle45] Stephen C. Kleene. On the interpretation of intuitionistic number theory. *The Journal of Symbolic Logic*, 10(4):109–124, 1945.

[Koz83] Dexter Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[Kre59] Georg Kreisel. Interpretation of analysis by means of functionals of finite type. In A. Heyting, editor, *Constructivity in Mathematics*, pages 101–128. North-Holland, 1959.

[Laf03] Y. Lafont. Towards an algebraic theory of boolean circuits. *Journal of Pure and Applied Algebra*, 184:257–310, 2003.

[LMW10] Y. Lafont, F. Métayer, and K. Worytkiewicz. A folk model structure on omega-cat. *Advances in Mathematics*, 224(3):1183–1231, 2010.

[Lur09] Jacob Lurie. *Higher topos theory*, volume 170 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ, 2009.

[Mal04] P. Malbos. *Critères de finitude homologique pour la non convergence des systèmes de réécriture de termes*. PhD thesis, Univ. Montpellier 2, 2004.

[Mé03] François Métayer. Resolutions by polygraphs. *Theory Appl. Categ.*, 11:148–184, 2003.

[OKK98] Friedrich Otto, Masashi Katsura, and Yuji Kobayashi. Infinite convergent string-rewriting systems and cross-sections for finitely presented monoids. *J. Symbolic Comput.*, 26(5):621–648, 1998.

[Pri70] Stewart B. Priddy. Koszul resolutions. *Trans. Amer. Math. Soc.*, 152:39–60, 1970.

[Pro13] The Univalent Foundations Program. *Homotopy type theory—univalent foundations of mathematics*. The Univalent Foundations Program, Princeton, NJ; Institute for Advanced Study (IAS), Princeton, NJ, 2013.

[PT18] Pierre-Marie Pédrot and Nicolas Tabareau. Failure is not an option - an exceptional type theory. In Amal Ahmed, editor, *Programming Languages and Systems - 27th European Symposium on Programming, ESOP 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings*, volume 10801 of *Lecture Notes in Computer Science*, pages 245–271. Springer, 2018.

[SOK94] Craig Squier, Friedrich Otto, and Yuji Kobayashi. A finiteness condition for rewriting systems. *Theoret. Comput. Sci.*, 131(2):271–294, 1994.

[Squ87] C. C. Squier. Word problems and a homological finiteness condition for monoids. *J. Pure Appl. Algebra*, 49(1-2):201–217, 1987.

[Ste04] Richard Steiner. Omega-categories and chain complexes. *Homology Homotopy Appl.*, 6(1):175–200, 2004.

[Str76] R. Street. Limits indexed by category-valued 2-functors. *Journal of Pure and Applied Algebra*, 8:149–181, 1976.

[Tit81] Jacques Tits. A local approach to buildings. In *The geometric vein*, pages 519–547. Springer, New York, 1981.