RESEARCH CENTRE

Saclay - Île-de-France

2020 ACTIVITY REPORT

Project-Team SPECFUN

Symbolic Special Functions : Fast and Certified

DOMAIN

Algorithmics, Programming, Software and Architecture

THEME

Algorithmics, Computer Algebra and Cryptology

Contents

Project-Team SPECFUN

Creation of the Team: 2012 November 01, updated into Project-Team: 2014 July 01

Keywords

Computer sciences and digital sciences

A2.1.11. – Proof languages

A2.4.3. – Proofs

A4.5. – Formal methods for security

A7.2. – Logic in Computer Science

A8.1. – Discrete mathematics, combinatorics

A8.3. – Geometry, Topology

A8.4. – Computer Algebra

A8.5. – Number theory

Other research topics and application domains

B9.5.2. – Mathematics

B9.5.3. – Physics

1 Team members, visitors, external collaborators

Research Scientists

- Frédéric Chyzak [Team leader, Inria, Researcher, HDR]
- Alin Bostan [Inria, Researcher, HDR]
- Guy Fayolle [Inria, Emeritus]
- Georges Gonthier [Inria, Senior Researcher, until Oct 2020]
- Pierre Lairez [Inria, Researcher]

Faculty Member

- Philippe Dumas [Ministère de l'Education Nationale]

PhD Students

- Alexandre Goyer [Inria, from Oct 2020]
- Rafael Mohr [Technische Universität Kaiserslautern (Germany), from Nov 2020]
- Raphaël Pagès [Université de Bordeaux, from Sep 2020]
- Sergey Yurkevich [University of Vienna (Austria), from Oct 2020]

Interns and Apprentices

- Alexandre Goyer [Sorbonne Université, from Mar 2020 until Jul 2020]
- Lucas Morisset [Inria, from Jun 2020 until Jul 2020]
- Raphaël Pagès [Université Paris Diderot, from Mar 2020 until Jul 2020]
- Nicholas Rumiz [Inria, from Mar 2020 until Jul 2020]

Administrative Assistant

- Bahar Carabetta [Inria]

Visiting Scientist

- Antonio Jiménez Pastor [Johannes Kepler University (Linz, Austria), from Feb 2020 until May 2020]

External Collaborator

- Marc Mezzarobba [CNRS]

2 Overall objectives

2.1 Scientific challenges, expected impact

The general orientation of our team is described by the short name given to it: *Special Functions*, that is, particular mathematical functions that have established names due to their importance in mathematical analysis, physics, and other application domains. Indeed, we ambition to study special functions with the computer, by combined means of computer algebra and formal methods.

Computer-algebra systems have been advertised for decades as software for "doing mathematics by computer" [87]. For instance, computer-algebra libraries can uniformly generate a corpus of mathematical properties about special functions, so as to display them on an interactive website. This possibility was recently shown by the computer-algebra component of the team [40]. Such an automated generation significantly increases the reliability of the mathematical corpus, in comparison to the content of existing static authoritative handbooks. The importance of the validity of these contents can be measured by the very wide audience that such handbooks have had, to the point that a book like [37] remains one of the most cited mathematical publications ever and has motivated the 10-year-long project of writing its successor [77]. However, can the mathematics produced "by computer" be considered as *true* mathematics? More specifically, whereas it is nowadays well established that the computer helps in discovering and observing new mathematical phenomenons, can the mathematical statements produced with the aid of the computer and the mathematical results computed by it be accepted as valid mathematics, that is, as having the status of mathematical *proofs*? Beyond the reported weaknesses or controversial design choices of mainstream computer-algebra systems, the issue is more of an epistemological nature. It will not find its solution even in the advent of the ultimate computer-algebra system: the social process of peer-reviewing just falls short of evaluating the results produced by computers, as reported by Th. Hales [64] after the publication of his proof of the Kepler Conjecture about sphere packing.

A natural answer to this deadlock is to move to an alternative kind of mathematical software and to use a proof assistant to check the correctness of the desired properties or formulas. The success of large-scale formalization projects, like the Four-Color Theorem of graph theory [59], the above-mentioned Kepler Conjecture [64], and the Odd Order Theorem of group theory [1], have increased the understanding of the appropriate software-engineering methods for this peculiar kind of programming. For computer algebra, this legitimates a move to proof assistants now.

The Dynamic Dictionary of Mathematical Functions [2] (DDMF) [40] is an online computer-generated handbook of mathematical functions that ambitions to serve as a reference for a broad range of applications. This software was developed by the computer-algebra component of the team as a project [3] of the MSR–INRIA Joint Centre. It bases on a library for the computer-algebra system Maple, Algolib [4], whose development started 20 years ago in project-team Algorithms [5]. As suggested by the constant questioning of certainty by new potential users, DDMF deserves a formal guarantee of correctness of its content, on a level that proof assistants can provide. Fortunately, the maturity of special-functions algorithms in Algolib makes DDMF a stepping stone for such a formalization: it provides a well-understood and unified algorithmic treatment, without which a formal certification would simply be unreachable.

The formal-proofs component of the team emanates from another project of the MSR–INRIA Joint Centre, namely the Mathematical Components project (MathComp) [6]. Since 2006, the MathComp group has endeavoured to develop computer-checked libraries of formalized mathematics, using the Coq proof assistant [83]. The methodological aim of the project was to understand the design methods leading to successful large-scale formalizations. The work culminated in 2012 with the completion of a formal proof of the Odd Order Theorem, resulting in the largest corpus of algebraic theories ever machine-checked with a proof assistant and a whole methodology to effectively combine these components in order to tackle complex formalizations. In particular, these libraries provide a good number of the many algebraic

[1] http://www.msr-inria.inria.fr/news/the-formalization-of-the-odd-order-theorem-has-been-completed-the-20-septembre-2012/
[2] http://ddmf.msr-inria.inria.fr/1.9.1/ddmf
[3] http://www.msr-inria.inria.fr/projects/dynamic-dictionary-of-mathematical-functions/
[4] http://algo.inria.fr/libraries/
[5] http://algo.inria.fr/
[6] http://www.msr-inria.fr/projects/mathematical-components/

objects needed to reason about special functions and their properties, like rational numbers, iterated sums, polynomials, and a rich hierarchy of algebraic structures.

The present team takes benefit from these recent advances to explore the formal certification of the results collected in DDMF. The aim of this project is to concentrate the formalization effort on this delimited area, building on DDMF and the Algolib library, as well as on the Coq system [83] and on the libraries developed by the MathComp project.

2.2 Use computer algebra but convince users beyond reasonable doubt

The following few opinions on computer algebra are, we believe, typical of computer-algebra users' doubts and difficulties when using computer-algebra systems:

- Fredrik Johansson, expert in the multi-precision numerical evaluation of special functions and in fast computer-algebra algorithms, writes on his blog [70]: "Mathematica is great for cross-checking numerical values, but it's not unusual to run into bugs, so *triple checking is a good habit*." One answer in the discussion is: "We can claim that Mathematica has [...] *an impossible to understand semantics*: If Mathematica's output is wrong then change the input. If you don't like the answer, change the question. That seems to be the philosophy behind."

- A professor's advice to students [79] on using Maple: "You may wish to use Maple to check your homework answers. If you do then keep in mind that Maple sometimes gives the *wrong answer, usually because you asked incorrectly, or because of niceties of analytic continuation*. You may even be bitten by an occasional Maple bug, though that has become fairly unlikely. Even with as powerful a tool as Maple you will still *have to devise your own checks* and you will still have to think."

- Jacques Carette, former head of the maths group at Maplesoft, about a bug [73] when asking Maple to take the limit `limit(f(n) * exp(-n), n = infinity)` for an undetermined function `f`: "The problem is that there is an *implicit assumption in the implementation* that unknown functions do not 'grow too fast'."

As explained by the expert views above, complaints by computer-algebra users are often due to their misunderstanding of what a computer-algebra systems is, namely a purely syntactic tool for calculations, that the user must complement with a semantics. Still, robustness and consistency of computer-algebra systems are not ensured as of today, and, whatever Zeilberger may provocatively say in his Opinion 94 [89], a firmer logical foundation is necessary. Indeed, the fact is that many "bugs" in a computer-algebra system cannot be fixed by just the usual debugging method of tracking down the faulty lines in the code. It is sort of "by design": assumptions that too often remain implicit are really needed by the design of symbolic algorithms and cannot easily be expressed in the programming languages used in computer algebra. A similar certification initiative has already been undertaken in the domain of numerical computing, in a successful manner [66, 43]. It is natural to undertake a similar approach for computer algebra.

2.3 Make computer algebra and formal proofs help one another

Some of the mathematical objects that interest our team are still totally untouched by formalization. When implementing them and their theory inside a proof assistant, we have to deal with the pervasive discrepancy between the published literature and the actual implementation of computer-algebra algorithms. Interestingly, this forces us to clarify our computer-algebraic view on them, and possibly make us discover holes lurking in published (human) proofs. We are therefore convinced that the close interaction of researchers from both fields, which is what we strive to maintain in this team, is a strong asset.

For a concrete example, the core of Zeilberger's creative telescoping manipulates rational functions up to simplifications. In summation applications, checking that these simplifications do not hide problematic divisions by 0 is most often left to the reader. In the same vein, in the case of integrals, the published algorithms do not check the convergence of all integrals, especially in intermediate calculations. Such checks are again left to the readers. In general, we expect to revisit the existing algorithms to ensure that they are meaningful for genuine mathematical sequences or functions, and not only for algebraic idealizations.

Another big challenge in this project originates in the scientific difference between computer algebra and formal proofs. Computer algebra seeks speed of calculation on *concrete instances* of algebraic data structures (polynomials, matrices, etc). For their part, formal proofs manipulate symbolic expressions in terms of *abstract variables* understood to represent generic elements of algebraic data structures. In view of this, a continuous challenge is to develop the right, hybrid thinking attitude that is able to effectively manage concrete and abstract values simultaneously, alternatively computing and proving with them.

2.4 Experimental mathematics with special functions

Applications in combinatorics and mathematical physics frequently involve equations of so high orders and so large sizes, that computing or even storing all their coefficients is impossible on existing computers. Making this tractable is an extraordinary challenge. The approach we believe in is to design algorithms of good—ideally quasi-optimal—complexity in order to extract precisely the required data from the equations, while avoiding the computationally intractable task of completely expanding them into an explicit representation.

Typical applications with expected high impact are the automatic discovery and algorithmic proof of results in combinatorics and mathematical physics for which human proofs are currently unattainable.

2.5 Research axes

The implementation of certified symbolic computations on special functions in the Coq proof assistant requires both investigating new formalization techniques and renewing the traditional computer-algebra viewpoint on these standard objects. Large mathematical objects typical of computer algebra occur during formalization, which also requires us to improve the efficiency and ergonomics of Coq. In order to feed this interdisciplinary activity with new motivating problems, we additionally pursue a research activity oriented towards experimental mathematics in application domains that involve special functions. We expect these applications to pose new algorithmic challenges to computer algebra, which in turn will deserve a formal-certification effort. Finally, DDMF is the motivation and the showcase of our progress on the certification of these computations. While striving to provide a formal guarantee of the correctness of the information it displays, we remain keen on enriching its mathematical content by developing new computer-algebra algorithms.

2.6 Computer algebra certified by the Coq system

Our formalization effort consists in organizing a cooperation between a computer-algebra system and a proof assistant. The computer-algebra system is used to produce efficiently algebraic data, which are later processed by the proof assistant. The success of this cooperation relies on the design of appropriate libraries of formalized mathematics, including certified implementations of certain computer-algebra algorithms. On the other side, we expect that scrutinizing the implementation and the output of computer-algebra algorithms will shed a new light on their semantics and on their correctness proofs, and help clarifying their documentation.

2.6.1 Libraries of formalized mathematics

The appropriate framework for the study of efficient algorithms for special functions is *algebraic*. Representing algebraic theories as Coq formal libraries takes benefit from the methodology emerging from the success of ambitious projects like the formal proof of a major classification result in finite-group theory (the Odd Order Theorem) [57].

Yet, a number of the objects we need to formalize in the present context has never been investigated using any interactive proof assistant, despite being considered as commonplaces in computer algebra. For instance there is up to our knowledge no available formalization of the theory of non-commutative rings, of the algorithmic theory of special-functions closures, or of the asymptotic study of special functions. We expect our future formal libraries to prove broadly reusable in later formalizations of seemingly unrelated theories.

2.6.2 Manipulation of large algebraic data in a proof assistant

Another peculiarity of the mathematical objects we are going to manipulate with the Coq system is their size. In order to provide a formal guarantee on the data displayed by DDMF, two related axes of research have to be pursued. First, efficient algorithms dealing with these large objects have to be programmed and run in Coq. Recent evolutions of the Coq system to improve the efficiency of its internal computations [38, 41] make this objective reachable. Still, how to combine the aforementioned formalization methodology with these cutting-edge evolutions of Coq remains one of the prospective aspects of our project. A second need is to help users *interactively* manipulate large expressions occurring in their conjectures, an objective for which little has been done so far. To address this need, we work on improving the ergonomics of the system in two ways: first, ameliorating the reactivity of Coq in its interaction with the user; second, designing and implementing extensions of its interface to ease our formalization activity. We expect the outcome of these lines of research to be useful to a wider audience, interested in manipulating large formulas on topics possibly unrelated to special functions.

2.6.3 Formal-proof-producing normalization algorithms

Our algorithm certifications inside Coq intend to simulate well-identified components of our Maple packages, possibly by reproducing them in Coq. It would however not have been judicious to re-implement them inside Coq in a systematic way. Indeed for a number of its components, the output of the algorithm is more easily checked than found, like for instance the solving of a linear system. Rather, we delegate the discovery of the solutions to an external, untrusted oracle like Maple. Trusted computations inside Coq then formally validate the correctness of the a priori untrusted output. More often than not, this validation consists in implementing and executing normalization procedures *inside* Coq. A challenge of this automation is to make sure they go to scale while remaining efficient, which requires a Coq version of non-trivial computer-algebra algorithms. A first, archetypal example we expect to work on is a non-commutative generalization of the normalization procedure for elements of rings [63].

2.7 Better symbolic computations with special functions

Generally speaking, we design algorithms for manipulating special functions symbolically, whether univariate or with parameters, and for extracting algorithmically any kind of algebraic and analytic information from them, notably asymptotic properties. Beyond this, the heart of our research is concerned with parametrised definite summations and integrations. These very expressive operations have far-ranging applications, for instance, to the computation of integral transforms (Laplace, Fourier) or to the solution of combinatorial problems expressed via integrals (coefficient extractions, diagonals). The algorithms that we design for them need to really operate on the level of linear functional systems, differential and of recurrence. In all cases, we strive to design our algorithms with the constant goal of good theoretical complexity, and we observe that our algorithms are also fast in practice.

2.7.1 Special-function integration and summation

Our long-term goal is to design fast algorithms for a general method for special-function integration (*creative telescoping*), and make them applicable to general special-function inputs. Still, our strategy is to proceed with simpler, more specific classes first (rational functions, then algebraic functions, hyperexponential functions, D-finite functions, non-D-finite functions; two variables, then many variables); as well, we isolate analytic questions by first considering types of integration with a more purely algebraic flavor (constant terms, algebraic residues, diagonals of combinatorics). In particular, we expect to extend our recent approach [46] to more general classes (algebraic with nested radicals, for example): the idea is to speed up calculations by making use of an analogue of Hermite reduction that avoids considering certificates. Homologous problems for summation will be addressed as well.

2.7.2 Applications to experimental mathematics

As a consequence of our complexity-driven approach to algorithms design, the algorithms mentioned in the previous paragraph are of good complexity. Therefore, they naturally help us deal with applications

that involve equations of high orders and large sizes.

With regard to combinatorics, we expect to advance the algorithmic classification of combinatorial classes like walks and urns. Here, the goal is to determine if enumerative generating functions are rational, algebraic, or D-finite, for example. Physical problems whose modelling involves special-function integrals comprise the study of models of statistical mechanics, like the Ising model for ferro-magnetism, or questions related to Hamiltonian systems.

Number theory is another promising domain of applications. Here, we attempt an experimental approach to the automated certification of integrality of the coefficients of mirror maps for Calabi–Yau manifolds. This could also involve the discovery of new Calabi–Yau operators and the certification of the existing ones. We also plan to algorithmically discover and certify new recurrences yielding good approximants needed in irrationality proofs.

It is to be noted that in all of these application domains, we would so far use general algorithms, as was done in earlier works of ours [45, 49, 47]. To push the scale of applications further, we plan to consider in each case the specifics of the application domain to tailor our algorithms.

2.8 Interactive and certified mathematical web sites

In continuation of our past project of an encyclopedia at `http://ddmf.msr-inria.inria.fr/1.9.1/ddmf`, we ambition to both enrich and certify the formulas about the special functions that we provide online. For each function, our website shows its essential properties and the mathematical objects attached to it, which are often infinite in nature (numerical evaluations, asymptotic expansions). An interactive presentation has the advantage of allowing for adaption to the user's needs. More advanced content will broaden the encyclopedia:

- the algorithmic discussion of equations with parameters, leading to certified automatic case analysis based on arithmetic properties of the parameters;
- lists of summation and integral formulas involving special functions, including validity conditions on the parameters;
- guaranteed large-precision numerical evaluations.

3 Research program

3.1 Studying special functions by computer algebra

Computer algebra manipulates symbolic representations of exact mathematical objects in a computer, in order to perform computations and operations like simplifying expressions and solving equations for "closed-form expressions". The manipulations are often fundamentally of algebraic nature, even when the ultimate goal is analytic. The issue of efficiency is a particular one in computer algebra, owing to the extreme swell of the intermediate values during calculations.

Our view on the domain is that research on the algorithmic manipulation of special functions is anchored between two paradigms:

- adopting linear differential equations as the right data structure for special functions,
- designing efficient algorithms in a complexity-driven way.

It aims at four kinds of algorithmic goals:

- algorithms combining functions,
- functional equations solving,
- multi-precision numerical evaluations,
- guessing heuristics.

This interacts with three domains of research:

- computer algebra, meant as the search for quasi-optimal algorithms for exact algebraic objects,
- symbolic analysis/algebraic analysis;
- experimental mathematics (combinatorics, mathematical physics, ...).

This view is made explicit in the present section.

3.2 Equations as a data structure

Numerous special functions satisfy linear differential and/or recurrence equations. Under a mild technical condition, the existence of such equations induces a finiteness property that makes the main properties of the functions decidable. We thus speak of *D-finite functions*. For example, 60 % of the chapters in the handbook [37] describe D-finite functions. In addition, the class is closed under a rich set of algebraic operations. This makes linear functional equations just the right data structure to encode and manipulate special functions. The power of this representation was observed in the early 1990s [88], leading to the design of many algorithms in computer algebra. Both on the theoretical and algorithmic sides, the study of D-finite functions shares much with neighbouring mathematical domains: differential algebra, D-module theory, differential Galois theory, as well as their counterparts for recurrence equations.

3.3 Algorithms combining functions

Differential/recurrence equations that define special functions can be recombined [88] to define: additions and products of special functions; compositions of special functions; integrals and sums involving special functions. Zeilberger's fast algorithm for obtaining recurrences satisfied by parametrised binomial sums was developed in the early 1990s already [90]. It is the basis of all modern definite summation and integration algorithms. The theory was made fully rigorous and algorithmic in later works, mostly by a group in RISC (Linz, Austria) and by members of the team [78, 86, 52, 50, 51, 71]. The past ÉPI Algorithms contributed several implementations (gfun [81], Mgfun [52]).

3.4 Solving functional equations

Encoding special functions as defining linear functional equations postpones some of the difficulty of the problems to a delayed solving of equations. But at the same time, solving (for special classes of functions) is a sub-task of many algorithms on special functions, especially so when solving in terms of polynomial or rational functions. A lot of work has been done in this direction in the 1990s; more intensively since the 2000s, solving differential and recurrence equations in terms of special functions has also been investigated.

3.5 Multi-precision numerical evaluation

A major conceptual and algorithmic difference exists for numerical calculations between data structures that fit on a machine word and data structures of arbitrary length, that is, *multi-precision* arithmetic. When multi-precision floating-point numbers became available, early works on the evaluation of special functions were just promising that "most" digits in the output were correct, and performed by heuristically increasing precision during intermediate calculations, without intended rigour. The original theory has evolved in a twofold way since the 1990s: by making computable all constants hidden in asymptotic approximations, it became possible to guarantee a *prescribed* absolute precision; by employing state-of-the-art algorithms on polynomials, matrices, etc, it became possible to have evaluation algorithms in a time complexity that is linear in the output size, with a constant that is not more than a few units. On the implementation side, several original works exist, one of which (NumGfun [76]) is used in our DDMF.

3.6 Guessing heuristics

"Differential approximation", or "Guessing", is an operation to get an ODE likely to be satisfied by a given approximate series expansion of an unknown function. This has been used at least since the 1970s and

is a key stone in spectacular applications in experimental mathematics [49]. All this is based on subtle algorithms for Hermite–Padé approximants [39]. Moreover, guessing can at times be complemented by proven quantitative results that turn the heuristics into an algorithm [48]. This is a promising algorithmic approach that deserves more attention than it has received so far.

3.7 Complexity-driven design of algorithms

The main concern of computer algebra has long been to prove the feasibility of a given problem, that is, to show the existence of an algorithmic solution for it. However, with the advent of faster and faster computers, complexity results have ceased to be of theoretical interest only. Nowadays, a large track of works in computer algebra is interested in developing fast algorithms, with time complexity as close as possible to linear in their output size. After most of the more pervasive objects like integers, polynomials, and matrices have been endowed with fast algorithms for the main operations on them [58], the community, including ourselves, started to turn its attention to differential and recurrence objects in the 2000s. The subject is still not as developed as in the commutative case, and a major challenge remains to understand the combinatorics behind summation and integration. On the methodological side, several paradigms occur repeatedly in fast algorithms: "divide and conquer" to balance calculations, "evaluation and interpolation" to avoid intermediate swell of data, etc. [44].

3.8 Encyclopedias

Handbooks collecting mathematical properties aim at serving as reference, therefore trusted, documents. The decision of several authors or maintainers of such knowledge bases to move from paper books [37, 77, 82] to websites and wikis [7] allows for a more collaborative effort in proof reading. Another step toward further confidence is to manage to generate the content of an encyclopedia by computer-algebra programs, as is the case with the Wolfram Functions Site [8] or DDMF [9]. Yet, due to the lingering doubts about computer-algebra systems, some encyclopedias propose both cross-checking by different systems and handwritten companion paper proofs of their content [10]. As of today, there is no encyclopedia certified with formal proofs.

3.9 Computer algebra and symbolic logic

Several attempts have been made in order to extend existing computer-algebra systems with symbolic manipulations of logical formulas. Yet, these works are more about extending the expressivity of computer-algebra systems than about improving the standards of correctness and semantics of the systems. Conversely, several projects have addressed the communication of a proof system with a computer-algebra system, resulting in an increased automation available in the proof system, to the price of the uncertainty of the computations performed by this oracle.

3.10 Certifying systems for computer algebra

More ambitious projects have tried to design a new computer-algebra system providing an environment where the user could both program efficiently and elaborate formal and machine-checked proofs of correctness, by calling a general-purpose proof assistant like the Coq system. This approach requires a huge manpower and a daunting effort in order to re-implement a complete computer-algebra system, as well as the libraries of formal mathematics required by such formal proofs.

3.11 Semantics for computer algebra

The move to machine-checked proofs of the mathematical correctness of the output of computer-algebra implementations demands a prior clarification about the often implicit assumptions on which the

[7] for instance http://dlmf.nist.gov/ for special functions or http://oeis.org/ for integer sequences
[8] http://functions.wolfram.com/
[9] http://ddmf.msr-inria.inria.fr/1.9.1/ddmf
[10] http://129.81.170.14/~vhm/Table.html

presumably correctly implemented algorithms rely. Interestingly, this preliminary work, which could be considered as independent from a formal certification project, is seldom precise or even available in the literature.

3.12 Formal proofs for symbolic components of computer-algebra systems

A number of authors have investigated ways to organize the communication of a chosen computer-algebra system with a chosen proof assistant in order to certify specific components of the computer-algebra systems, experimenting various combinations of systems and various formats for mathematical exchanges. Another line of research consists in the implementation and certification of computer-algebra algorithms inside the logic [85, 63, 72] or as a proof-automation strategy. Normalization algorithms are of special interest when they allow to check results possibly obtained by an external computer-algebra oracle [55]. A discussion about the systematic separation of the search for a solution and the checking of the solution is already clearly outlined in [69].

3.13 Formal proofs for numerical components of computer-algebra systems

Significant progress has been made in the certification of numerical applications by formal proofs. Libraries formalizing and implementing floating-point arithmetic as well as large numbers and arbitrary-precision arithmetic are available. These libraries are used to certify floating-point programs, implementations of mathematical functions and for applications like hybrid systems.

3.14 Machine-checked proofs of formalized mathematics

To be checked by a machine, a proof needs to be expressed in a constrained, relatively simple formal language. Proof assistants provide facilities to write proofs in such languages. But, as merely writing, even in a formal language, does not constitute a formal proof just per se, proof assistants also provide a proof checker: a small and well-understood piece of software in charge of verifying the correctness of arbitrarily large proofs. The gap between the low-level formal language a machine can check and the sophistication of an average page of mathematics is conspicuous and unavoidable. Proof assistants try to bridge this gap by offering facilities, like notations or automation, to support convenient formalization methodologies. Indeed, many aspects, from the logical foundation to the user interface, play an important role in the feasibility of formalized mathematics inside a proof assistant.

3.15 Logical foundations and proof assistants

While many logical foundations for mathematics have been proposed, studied, and implemented, type theory is the one that has been more successfully employed to formalize mathematics, to the notable exception of the Mizar system [74], which is based on set theory. In particular, the calculus of construction (CoC) [53] and its extension with inductive types (CIC) [54], have been studied for more than 20 years and been implemented by several independent tools (like Lego, Matita, and Agda). Its reference implementation, Coq [83], has been used for several large-scale formalizations projects (formal certification of a compiler back-end; four-color theorem). Improving the type theory underlying the Coq system remains an active area of research. Other systems based on different type theories do exist and, whilst being more oriented toward software verification, have been also used to verify results of mainstream mathematics (prime-number theorem; Kepler conjecture).

3.16 Computations in formal proofs

The most distinguishing feature of CoC is that computation is promoted to the status of rigorous logical argument. Moreover, in its extension CIC, we can recognize the key ingredients of a functional programming language like inductive types, pattern matching, and recursive functions. Indeed, one can program effectively inside tools based on CIC like Coq. This possibility has paved the way to many effective formalization techniques that were essential to the most impressive formalizations made in CIC.

Another milestone in the promotion of the computations-as-proofs feature of Coq has been the integration of compilation techniques in the system to speed up evaluation. Coq can now run realistic programs in the logic, and hence easily incorporates calculations into proofs that demand heavy computational steps.

Because of their different choice for the underlying logic, other proof assistants have to simulate computations outside the formal system, and indeed fewer attempts to formalize mathematical proofs involving heavy calculations have been made in these tools. The only notable exception, which was finished in 2014, the Kepler conjecture, required a significant work to optimize the rewriting engine that simulates evaluation in Isabelle/HOL.

3.17 Large-scale computations for proofs inside the Coq system

Programs run and proved correct inside the logic are especially useful for the conception of automated decision procedures. To this end, inductive types are used as an internal language for the description of mathematical objects by their syntax, thus enabling programs to reason and compute by case analysis and recursion on symbolic expressions.

The output of complex and optimized programs external to the proof assistant can also be stamped with a formal proof of correctness when their result is easier to *check* than to *find*. In that case one can benefit from their efficiency without compromising the level of confidence on their output at the price of writing and certify a checker inside the logic. This approach, which has been successfully used in various contexts, is very relevant to the present research project.

3.18 Relevant contributions from the Mathematical Component libraries

Representing abstract algebra in a proof assistant has been studied for long. The libraries developed by the MathComp project for the proof of the Odd Order Theorem provide a rather comprehensive hierarchy of structures; however, they originally feature a large number of instances of structures that they need to organize. On the methodological side, this hierarchy is an incarnation of an original work [57] based on various mechanisms, primarily type inference, typically employed in the area of programming languages. A large amount of information that is implicit in handwritten proofs, and that must become explicit at formalization time, can be systematically recovered following this methodology.

Small-scale reflection [60] is another methodology promoted by the MathComp project. Its ultimate goal is to ease formal proofs by systematically dealing with as many bureaucratic steps as possible, by automated computation. For instance, as opposed to the style advocated by Coq's standard library, decidable predicates are systematically represented using computable boolean functions: comparison on integers is expressed as program, and to state that $a \leq b$ one compares the output of this program run on a and b with $true$. In many cases, for example when a and b are values, one can prove or disprove the inequality by pure computation.

The MathComp library was consistently designed after uniform principles of software engineering. These principles range from simple ones, like naming conventions, to more advanced ones, like generic programming, resulting in a robust and reusable collection of formal mathematical components. This large body of formalized mathematics covers a broad panel of algebraic theories, including of course advanced topics of finite group theory, but also linear algebra, commutative algebra, Galois theory, and representation theory. We refer the interested reader to the online documentation of these libraries [84], which represent about 150,000 lines of code and include roughly 4,000 definitions and 13,000 theorems.

Topics not addressed by these libraries and that might be relevant to the present project include real analysis and differential equations. The most advanced work of formalization on these domains is available in the HOL-Light system [65, 67, 68], although some existing developments of interest [42, 75] are also available for Coq. Another aspect of the MathComp libraries that needs improvement, owing to the size of the data we manipulate, is the connection with efficient data structures and implementations, which only starts to be explored.

3.19 User interaction with the proof assistant

The user of a proof assistant describes the proof he wants to formalize in the system using a textual language. Depending on the peculiarities of the formal system and the applicative domain, different proof languages have been developed. Some proof assistants promote the use of a declarative language, when the Coq and Matita systems are more oriented toward a procedural style.

The development of the large, consistent body of MathComp libraries has prompted the need to design an alternative and coherent language extension for the Coq proof assistant [62, 61], enforcing the robustness of proof scripts to the numerous changes induced by code refactoring and enhancing the support for the methodology of small-scale reflection.

The development of large libraries is quite a novelty for the Coq system. In particular any long-term development process requires the iteration of many refactoring steps and very little support is provided by most proof assistants, with the notable exception of Mizar [80]. For the Coq system, this is an active area of research.

4 Application domains

4.1 Computer Algebra in Mathematics

Our expertise in computer algebra and complexity-driven design of algebraic algorithms has applications in various domains, including:

- combinatorics, especially the study of combinatorial walks,
- theoretical computer science, like by the study of automatic sequences,
- number theory, by the analysis of the nature of so-called periods.

5 New results

5.1 Algebraic algorithms on fundamental objects

5.1.1 A simple and fast algorithm for computing the N-th term of a linearly recurrent sequence

In [24] Alin Bostan and Ryuhei Mori (Tokyo Institute of Technology, Japan) designed a simple and fast algorithm for computing the N-th term of a given linearly recurrent sequence. The new algorithm uses $O(\mathsf{M}(d)\log N)$ arithmetic operations, where d is the order of the recurrence, and $\mathsf{M}(d)$ denotes the number of arithmetic operations for computing the product of two polynomials of degree d. The state-of-the-art algorithm, due to Fiduccia (1985), had the same arithmetic complexity up to a constant factor. The new algorithm is simpler, faster and obtained by a totally different method. They also discuss several algorithmic applications, notably to polynomial modular exponentiation ($P^N \bmod Q$), on which many other useful algorithms rely, either in computer algebra (e.g., polynomial factoring over finite fields), or in algorithmic number theory (e.g., primality tests) or in effective algebraic geometry (e.g., counting points on curves over finite fields).

5.1.2 Fast computation of the N-th term of a q-holonomic sequence and applications

In 1977, Strassen invented a famous baby-step/giant-step algorithm that computes the factorial $N!$ in arithmetic complexity quasi-linear in $\sqrt{N}$. In 1988, the Chudnovsky brothers generalized Strassen's algorithm to the computation of the N-th term of any holonomic sequence in essentially the same arithmetic complexity. In [22, 29], Alin Bostan together with his PhD student Sergey Yurkevich designed q-analogues of these algorithms. They first extend Strassen's algorithm to the computation of the q-factorial of N, then Chudnovskys' algorithm to the computation of the N-th term of any q-holonomic sequence. Both algorithms work in arithmetic complexity quasi-linear in $\sqrt{N}$; surprisingly, they are simpler than their analogues in the holonomic case. They provide a detailed cost analysis, in both arithmetic and bit complexity models. Moreover, they describe various algorithmic consequences,

including the acceleration of polynomial and rational solving of linear q-differential equations, and the fast evaluation of large classes of polynomials, including a family recently considered by Nogneng and Schost.

5.1.3 Subresultants of $(x-\alpha)^m$ and $(x-\beta)^n$, Jacobi polynomials and complexity

A previous article by Alin Bostan and colleagues last year described explicit expressions for the coefficients of the order-d polynomial subresultant of $(x-\alpha)^m$ and $(x-\beta)^n$ with respect to Bernstein's set of polynomials $\{(x-\alpha)^j(x-\beta)^{d-j}, 0 \le j \le d\}$, for $0 \le d < \min\{m,n\}$. In [17], Alin Bostan, together with T. Krick, M. Valdettaro (U. Buenos Aires, Argentina) and A. Szanto (U. North Carolina, Raleigh, USA) further developed the study of these structured polynomials and showed that the coefficients of the subresultants of $(x-\alpha)^m$ and $(x-\beta)^n$ with respect to the monomial basis can be computed in *linear* arithmetic complexity, which is faster than for arbitrary polynomials. The result is obtained as a consequence of the amazing though seemingly unnoticed fact that these subresultants are scalar multiples of Jacobi polynomials up to an affine change of variables.

5.1.4 Improved algorithms for left factorial residues

In [11], Alin Bostan together with Vladica Andrejić (University of Belgrade, Serbia) and Milos Tatarevic (CoinList, Alameda, CA) presented improved algorithms for computing the left factorial residues $!p = 0! + 1! + \cdots + (p-1)! \bmod p$. They used these algorithms for the calculation of the residues $!p \bmod p$, for all primes p up to 2^{40}. Their results confirm that Kurepa's left factorial conjecture is still an open problem, as they show that there are no odd primes $p < 2^{40}$ such that p divides $!p$. Additionally, they confirmed that there are no socialist primes p with $5 < p < 2^{40}$.

5.1.5 Explicit degree bounds for right factors of linear differential operators

If a linear differential operator with rational function coefficients is reducible, its factors may have coefficients with numerators and denominators of very high degree. When the base field is $\mathbb{C}$, Alin Bostan together with Bruno Salvy (Inria and ENS Lyon) and Tanguy Rivoal (CNRS and U. Grenoble) gave in [18] a completely explicit bound for the degrees of the monic right factors in terms of the degree and the order of the original operator, as well as the largest modulus of the local exponents at all its singularities. As a consequence, if a differential operator L has rational function coefficients over a number field, they obtain degree bounds for its monic right factors in terms of the degree, the order and the height of L, and of the degree of the number field.

5.2 Polynomial systems and nonlinear equations

5.2.1 Complexity analysis of structured polynomial system solving

With Peter Bürgisser (Technische Universität Berlin, Germany) and Felipe Cucker (City University of Hong Kong), Pierre Lairez has developed an algorithm to compute an approximate zero of a polynomial system given as a black-box evaluation function. Based on this, they study the average complexity of solving polynomial systems with low evaluation complexity.

Previous average complexity analyses of numerical algorithms to solve polynomial systems assume a dense model of random polynomials, far from the applications. In the work [31], Pierre Lairez and his colleagues deal with a model of random polynomials (random algebraic branching programs, ABP) indexed by the evaluation complexity and coming from a universal model of computation in algebraic complexity theory. They show that their algorithm performs $\mathrm{poly}(n,\delta)$ operations on average to solve random ABP in n variables, of degree δ and of size $\mathrm{poly}(n,\delta)$. This brings complexity analysis closer than ever to what happens in practice.

5.2.2 Separation of periods of quartic surfaces

Given two numbers A and B, a high precision numerical computation (say 10000 digits) can convince anybody that $A = B$ if the numerical approximations agree. However, this is not a proof. A separation

bound yields a theorem in the form "$A = B$ or $|A - B| > \epsilon$". With Emre Sertöz (formerly MPI Leipzig, Germany, now MPI Bonn), Pierre Lairez proved a separation bound for periods of quartic surfaces [35]. This is a surprising result as not much is known about general families of periods (although much is known about specific periods like π). Moreover, it gives credit to numerical approaches in effective algebraic geometry. Indeed, the usual criticism is the idea that numerical methods are good heuristics but cannot possibly prove anything. This result is a downright refutation of this argument.

5.2.3 A Gröbner-basis theory for divide-and-conquer recurrences

Divide-and-conquer recurrences relate values of a given sequence at indices of the form $b^\ell n + r$ for fixed $b \geq 2$ and various pairs (ℓ, r). Using systems of such recurrences makes it possible to express the behaviour of a quantity according to congruence classes modulo b^ℓ. The study of such systems is poorly developed so far. In particular, no consistency check is available and no theory of initial values has been developed yet. In an ongoing study in order to fill this gap, Frédéric Chyzak and Philippe Dumas have introduced a noncommutative multivariate polynomial setting to represent divide-and-conquer systems. This setting involves at the same time variables that behave like words in purely noncommutative algebras and variables governed by commutation rules like in skew polynomial extensions. In their work [25], they initiate the study of left ideals of such polynomials and they develop their Gröbner-basis theory, including the usual division and Buchberger algorithms. Strikingly, the nature of commutations generally prevents the leading monomial of a polynomial product to be the product of the leading monomials. To overcome the difficulty, they consider a specific monomial ordering, together with a restriction to monic divisors in intermediate steps. They also develop a variant of the F_4 algorithm with distinguishing features.

5.2.4 The Sage package *comb_walks* for walks in the quarter plane

With Antonio Jiménez-Pastor who was visiting the team during his doctoral preparation, Alin Bostan, Frédéric Chyzak, and Pierre Lairez worked on a new software library designed to work with generating functions that count walks in the quarter plane. With this library for the Sagemath system they offer a cohesive package that brings together all the required procedures for manipulating these generating functions, as well as a unified interface to deal with them. They also display on a public webpage results that this package computes. They presented their work as an extended abstract at the conference ISSAC [14].

5.2.5 Differential transcendence of Bell numbers and relatives: a Galois-theoretic approach

A power series is called *differentially transcendent* if it does not satisfy any algebraic differential equation. In 2003, Martin Klazar proved in an elementary but very clever way that the ordinary generating function of the famous combinatorial Bell numbers, counting partitions of sets, is differentially transcendent. In [27], Alin Bostan (of the team), together with Lucia Di Vizio (CNRS, Université de Versailles) and Kilian Raschel (CNRS, Université de Tours), showed that Klazar's result is an instance of a general phenomenon that can be proven in a compact way using difference Galois theory. In the paper, they present the main principles of this theory in order to prove a general result of differential transcendence, that they apply to many other (infinite classes of) examples of generating functions, including as very special cases the ones considered by Klazar. Most of their examples belong to Sheffer's class, well studied notably in umbral calculus. They all bring concrete evidence in support to the Pak-Yeliussizov conjecture, according to which a sequence whose both ordinary and exponential generating functions satisfy nonlinear differential equations with polynomial coefficients necessarily satisfies a *linear* recurrence with polynomial coefficients.

5.2.6 On the exponential generating function of labelled trees

In [16], Alin Bostan and Antonio Jiménez-Pastor (U. Linz, Austria), proved that the exponential generating function of labelled trees, $T(x) = \sum_{n\geq 1} \frac{n^{n-1}}{n!} x^n$, is not D$^\infty$-finite. In particular, this implies that, although $T(x)$ satisfies non-linear differential equations, these equations cannot be "too simple". In particular, $T(x)$ is not the quotient of two D-finite functions (satisfying linear differential equations with polynomial

coefficients), More generally, $T(x)$ does not satisfy any linear differential equation with D-finite coefficients. The proof ultimately relies on a result in differential Galois theory. Several open questions are left, of which one concerning the nature of the ordinary generating function of labelled trees, $\sum_{n\geq 1} n^{n-1}x^n$.

5.3 Applications to combinatorics and probability theory

5.3.1 Bijections between Łukasiewicz walks and generalized tandem walks

Frédéric Chyzak and Karen Yeats (University of Waterloo, Canada) have studied the enumeration by length of several walk models on the square lattice. In the work [32] published this year, they obtain bijections between walks in the upper half-plane returning to the x-axis walks in the quarter plane. A recent unpublished work by Bostan, Chyzak, and Mahboubi had given a bijection for models using small north, west, and south-east steps. In this year's publication, Chyzak and Yeats adapt and generalize it to a bijection between half-plane walks using those three steps in two colours and a quarter-plane model over the symmetrized step set consisting of north, north-west, west, south, south-east, and east. They then generalize their bijections to certain models with large steps: for given $p \geq 1$, a bijection is given between the half-plane and quarter-plane models obtained by keeping the small south-east step and replacing the two steps north and west of length 1 by the $p+1$ steps of length p in directions between north and west. This model is close to, but distinct from, the model of generalized tandem walks studied by Bousquet-Mélou, Fusy, and Raschel.

5.3.2 Weakly-unambiguous Parikh automata and their link to holonomic series

In [23] Alin Bostan together with Arnaud Carayol, Florent Koechlin and Cyril Nicaud (Univ. Marne-la-Vallée, France) investigated the connection between properties of formal languages and properties of their generating series, with a focus on the class of holonomic power series. They first proved a strong version of a conjecture by Castiglione and Massazza: weakly-unambiguous Parikh automata are equivalent to unambiguous two-way reversal bounded counter machines, and their multivariate generating series are holonomic. They then show that the converse is not true: they construct a language whose generating series is algebraic (thus holonomic), but which is inherently weakly-ambiguous as a Parikh automata language. Finally, they prove an effective decidability result for the inclusion problem for weakly-unambiguous Parikh automata, and provide an upper-bound on its complexity.

5.3.3 Counting walks with large steps in an orthant

In the past fifteen years, the enumeration of lattice walks with steps taken in a prescribed set and confined to a given cone, especially the first quadrant of the plane, has been intensely studied. As a result, the generating functions of quadrant walks are now well-understood, provided the allowed steps are *small*. In particular, having small steps is crucial for the definition of a certain group of bi-rational transformations of the plane. It has been proved that this group is finite if and only if the corresponding generating function is D-finite. This group is also the key to the uniform solution of 19 of the 23 small step models possessing a finite group. In contrast, almost nothing was known for walks with arbitrary steps. In [12], Alin Bostan together with Mireille Bousquet-Mélou (CNRS, Bordeaux) and Stephen Melczer (U. Pennsylvania, Philadelphia, USA), extended the definition of the group, or rather of the associated orbit, to this general case, and generalized the above uniform solution of small step models. When this approach works, it invariably yields a D-finite generating function. They applied it to many quadrant problems, including some infinite families. After developing the general theory, the authors of [12] considered the 13 110 two-dimensional models with steps in $\{-2,-1,0,1\}^2$ having at least one -2 coordinate. They proved that only 240 of them have a finite orbit, and solve 231 of them with their method. The 9 remaining models are the counterparts of the 4 models of the small step case that resist the uniform solution method (and which are known to have an algebraic generating function). They conjecture D-finiteness for their generating functions (but only two of them are likely to be algebraic!), and proved non-D-finiteness for the 12 870 models with an infinite orbit, except for 16 of them.

5.3.4 The generating function of Kreweras walks with interacting boundaries is not algebraic

Beaton, Owczarek and Xu (2019) studied generating functions of Kreweras walks and of reverse Kreweras walks in the quarter plane, with interacting boundaries. They proved that for the reverse Kreweras step set, the generating function is always algebraic, and for the Kreweras step set, the generating function is always D-finite. However, apart from the particular case where the interactions are symmetric in x and y, they left open the question of whether the latter one is algebraic. Using computer algebra tools, Alin Bostan, together with Manuel Kauers and Thibaut Verron (University, Linz, Austria) confirmed [28] the previous intuition that the generating function of Kreweras walks is not algebraic, apart from the particular case already identified.

5.3.5 Martin boundary of killed random walks on isoradial graphs

Alin Bostan contributed to an article by C. Boutillier (Sorbonne Université) and K. Raschel (CNRS, Université de Tours) [19], devoted to the study of random walks on isoradial graphs. Contrary to the lattice case, isoradial graphs are not translation invariant, do not admit any group structure and are spatially non-homogeneous. However, Boutillier and Raschel have been able to obtain analogues of a celebrated result by Ney and Spitzer (1966) on the so-called *Martin kernel* (ratio of Green functions started at different points). Alin Bostan provided in the Appendix two different proofs of the fact that some algebraic power series arising in this context have non-negative coefficients.

5.3.6 Stieltjes moment sequences for pattern-avoiding permutations

A small subset of combinatorial sequences have coefficients that can be represented as moments of a nonnegative measure on $[0,\infty)$. Such sequences are known as *Stieltjes moment sequences*. They have a number of useful properties, such as log-convexity, which in turn enables one to rigorously bound their growth constant from below.

In [15], Alin Bostan together with Andrew Elvey Price (Université de Bordeaux), Anthony Guttmann (University of Melbourne), and Jean-Marie Maillard (Sorbonne Université), studied some classical sequences in enumerative combinatorics, denoted $Av(\mathscr{P})$, and counting permutations of $\{1,2,\dots,n\}$ that avoid some given pattern $\mathscr{P}$. For increasing patterns $\mathscr{P} = (12\dots k)$, they showed that the corresponding sequences, $Av(123\dots k)$, are Stieltjes moment sequences, and explicitly determined the underlying density function, either exactly or numerically, by using the Stieltjes inversion formula as a fundamental tool.

They showed that the densities for $Av(1234)$ and $Av(12345)$, correspond to an order-one linear differential operator acting on a classical modular form given as a pullback of a Gaussian ${}_2F_1$ hypergeometric function, respectively to an order-two linear differential operator acting on the square of a classical modular form given as a pullback of a ${}_2F_1$ hypergeometric function. Moreover, these density functions are closely, but non-trivially, related to the density attached to the distance traveled by a walk in the plane with $k-1$ unit steps in random directions.

As a bonus, they studied the challenging case of the $Av(1324)$ sequence and gave compelling numerical evidence that this too is a Stieltjes moment sequence. Accepting this, they proved new lower bounds on the growth constant of this sequence, which are stronger than existing bounds. A further unproven assumption leads to even better bounds, which can be extrapolated to give a good estimate of the (unknown) growth constant.

5.3.7 A note on gamma triangles and local gamma vectors

Alin Bostan contributed to F. Chapoton's article [20] by writing an appendix, which allowed the author to complete its article. The theme of [20] is the study of simplicial complexes in algebraic combinatorics. A basic invariant is the f-vector that counts faces according to their dimensions. A less understood invariant is the γ-vector, introduced by Gal in 2005. Also in 2005, Chapoton, motivated by the study of the combinatorics of simplicial complexes attached to cluster algebras, considered a refined version of the f-vector. The main aim of [20] is to introduce the analogue in this context of the γ-vector, and a further refinement called the Γ-triangle. The author computed explicitly the Γ-triangle for all the cluster

simplicial complexes of irreducible Coxeter groups. Alin Bostan contributed to the proof of an unexpected relation between the Γ-triangles of cluster fans of type $\mathbb{B}$ and $\mathbb{D}$.

5.3.8 On an integral identity

In [13], Alin Bostan together with Fernando Chamizo (Universidad Autónoma de Madrid and ICMAT, Spain) and Mikael Persson Sundqvist (Lund University, Sweden) gave three elementary proofs of a nice equality of definite integrals, recently proven by Ekhad, Zeilberger and Zudilin. The equality arises in the theory of bivariate hypergeometric functions, and has connections with irrationality proofs in number theory. They furthermore provide a generalization together with an equally elementary proof and discuss some consequences.

5.3.9 Diagonal representation of algebraic power series: a glimpse behind the scenes

There are many viewpoints on algebraic power series, ranging from the abstract ring-theoretic notion of Henselization to the very explicit perspective as diagonals of certain rational functions. Denef and Lipshitz proved in 1987 that any algebraic power series in n variables can be written as a diagonal of a rational power series in one variable more. Their proof uses a lot of involved theory and machinery which remains hidden to the reader in the original article. In the work [26], which is based on his master's thesis, Sergey Yurkevich explained these tools by motivating while defining them and reproving most of their interesting parts. Moreover, he provided a new significant improvement on the Artin-Mazur lemma, proving the existence of a 2-dimensional code of algebraic power series.

5.3.10 On a class of hypergeometric diagonals

In [30], Alin Bostan together with his PhD student Sergey Yurkevich proved that the diagonal of any finite product of algebraic functions of the form

$$(1 - x_1 - \ldots - x_n)^R, \qquad R \in \mathbb{Q},$$

is a generalized hypergeometric function, and they provided explicit description of its parameters. The particular case $(1-x-y)^R/(1-x-y-z)$ corresponds to the main identity of Abdelaziz, Koutschan and Maillard in [36, §3.2]. The result in [30] is useful in both directions: on the one hand it shows that Christol's conjecture holds true for a large class of hypergeometric functions, on the other hand it allows for a very explicit and general viewpoint on the diagonals of algebraic functions of the type above. Finally, in contrast to [36], the new proof is completely elementary and does not require any algorithmic help.

5.3.11 Genus and classification of random walks in the quarter plane

In collaboration with R. Iasnogorodski (SPCPA, Saint-Petersburg), Guy Fayolle analyzes the *kernel* $K(x,y,t)$ of the basic functional equation associated with the tri-variate counting generating function (CGF) of walks in the quarter plane. In this short paper [21], taking $t \in]0,1[$, we provide the conditions on the step set $\{p_{i,j}\}$ to decide whether the walks are *singular* or *regular*, as defined in [56, Section 2.3]. These conditions are independent of $t \in]0,1[$ and given in terms of *step set configurations*. They also find the configurations for the kernel to be of genus 0, knowing that the genus is always ≤ 1. All these conditions are very similar to the case $t = 1$ considered in [56]. Their results extend the work https://arxiv.org/abs/2004.01035, which considers only very special situations, namely when $t \in]0,1[$ is a transcendental number over the field $Q(p_{i,j})$.

5.3.12 Reflected Brownian motion in a nonconvex cone

In an ongoing work in collaboration with S. Franceschi (LMO, Paris-Saclay University) and K. Raschel (CNRS, Tours University), Guy Fayolle states a system of functional equations satisfied by the Laplace transform of the stationary distribution of a reflected Brownian motion (SRBM) in a two-dimensional non-convex cone. While the case of convex cones is now reasonably well studied, the framework of non-convex cones turns out to be more challenging, as shown by similar research carried out in a discrete

setting. They show in particular that the problem can be reduced to a boundary value problem of Rieman–Hilbert–Carleman type on an hyperbola, for a two-dimensional vector of meromorphic functions. This seems to be a quite original result.

5.3.13 Random walks in orthants and lattice path combinatorics

In the second edition of the book [56], original methods were proposed to determine the invariant measure of random walks in the quarter plane with small jumps (size 1), the general solution being obtained via reduction to boundary value problems. Among other things, an important quantity, the so-called *group of the walk*, allows to deduce theoretical features about the nature of the solutions. In particular, when the *order* of the group is finite and the underlying algebraic curve is of genus 0 or 1, necessary and sufficient conditions have been given for the solution to be rational, algebraic or D-finite (i.e. solution of a linear differential equation). In this framework, a number of difficult open problems related to lattice path combinatorics are currently being explored boundary Alin Bostan, Frédéric Chyzak, and Guy Fayolle, both from theoretical and computer algebra points of view: concrete computation of the criteria, utilization of differential Galois theory, genus greater than 1 (i.e., when some jumps are of size ≥ 2), etc. This relates simple product-form stochastic networks (so-called *Jackson networks*) and explicit solutions of functional equations for counting lattice walks. Some partial extensions of [33] are under development.

5.4 Formal methods

5.4.1 Recursive operator definitions

TLA+ originally allowed recursive function definitions, but not recursive operator definitions, because it was not known how how to define their semantics. They were added to the language in 2006 after a semantics was discovered for them. This year, Georges Gonthier, together with Leslie Lamport (Microsoft Research, USA), described that semantics in [34].

6 Partnerships and cooperations

6.1 International research visitors

6.1.1 Visits to international teams

Research stays abroad

- Alin Bostan visited (March 2020) Herwig Hauser's team at the University of Vienna. The initial plan was that Alin Bostan would be invited professor from March to June 2020, and that he would teach a Master course titled "Experimental Mathematics and Computer Algebra for Combinatorics and Number Theory". Unfortunately, his research stay and his lectures had to be aborted after few weeks due to the COVID-19 pandemic, and postponed.

6.2 National initiatives

6.2.1 ANR

- *De rerum natura*. This project, set up by the team, was accepted this year and will be funded until 2023. It gathers over 20 experts from four fields: computer algebra; the Galois theories of linear functional equations; number theory; combinatorics and probability. Our goal is to obtain classification algorithms for number theory and combinatorics, particularly so for deciding irrationality and transcendence. (Permanent members with pm listed: Bostan, Chyzak, Lairez.)

- *∂ifference*. This project, led by Olivier Bournez (Lix), started in November 2020. Its objective is to consider a novel approach in between the two worlds: discrete-oriented computations on the one side and differential equations on the other side. We aims at providing new insights on classical complexity theory, computability and logic through this prism and at introducing

new perspectives in algorithmic methods for differential equations solving and computer science applications. (Permanent members with pm listed: Bostan, Chyzak.)

- *Tremplin ERC*. Pierre Lairez has been awarded a "tremplin" project by ANR. This will help him prepare an ERC project submission "10000 Digits, Foundations of transcendental methods in numerical algebraic geometry".

6.3 Regional initiatives

- Alin Bostan submitted in November 2020 a PCRI-ANR proposal ELEFANT – "Efficient aLgorithms For guessing, summAtioN and posiTivity". This is a bilateral ANR/FWF project between 2 computer algebra teams in France and 2 computer algebra teams in Austria. The Austrian co-leader is Manuel Kauers from Univ. Linz. The goal is to work together on four axes: structured and multivariate guessing, positivity and D-finiteness, creative telescoping and applications in combinatorics, number theory and theoretical physics. The requested funding is of 770,000 euros in total.

- Alin Bostan is co-leader of the bilateral project "Integer Sequences arising in Number Theory, Combinatorics and Physics" between France and Austria. The Austrian co-leader is Herwig Hauser (U. Vienna, Austria).

7 Dissemination

7.1 Promoting scientific activities

7.1.1 Scientific events: organisation

General chair, scientific chair

- Alin Bostan is part of the Scientific advisory board of the conference series *Effective Methods in Algebraic Geometry* (MEGA).

- Since 2020, for a period of 5 years, Alin Bostan is member of the steering committee of the *Journées Nationales de Calcul Formel* (JNCF), the annual meeting of the French computer algebra community.

- Alin Bostan is part of the scientific committee of the GDR EFI ("Functional Equations and Interactions") dependent on the mathematical institute (INSMI) of the CNRS. The goal of this GDR is to bring together various research communities in France working on functional equations in fields of computer science and mathematics.

- Frédéric Chyzak is General Chair of the international conference ISSAC'21 (International Symposium on Symbolic and Algebraic Computation), whose organizing started in 2020.

Member of the organizing committees

- Alin Bostan co-organizes, with Lucia Di Vizio, the *Séminaire Différentiel* between U. Versailles and Inria Saclay, with a bi-annual frequency.

- Alin Bostan co-organizes, with Lucia Di Vizio and Kilian Raschel the working group *Transcendance et Combinatoire*, at Institut Henri Poincaré (Paris), with a bi-monthly frequency.

- Alin Bostan, together with Mohab Safey El Din, Bruno Salvy and Gilles Villard, wrote a proposal for a thematic program "Recent Trends in Computer Algebra (RTCA)", to be held in 2023 in Paris and Lyon. The proposal has been accepted, the main funders being IHP (120,000 euros) and Labex Milyon (60,000 euros).

7.1.2 Scientific events: selection

Member of the conference program committees

- Pierre Lairez has served in the program committee of ISSAC'20 (International Symposium on Symbolic and Algebraic Computation).

Reviewer

- Frédéric Chyzak has been a reviewer for *ISSAC'20 (International Symposium on Symbolic and Algebraic Computation)*, *CASC'20 (Computer Algebra in Scientific Computing)*, and for a post-proceeding of *Transient Transcendence in Transylvania*.
- Alin Bostan has been a reviewer for *ISSAC'20 (International Symposium on Symbolic and Algebraic Computation)* and *AofA'20 (Analysis of Algorithms)*.

7.1.3 Journal

Member of the editorial boards

- Alin Bostan is on the editorial board of the *Journal of Symbolic Computation*.
- Alin Bostan is on the editorial board of the *Annals of Combinatorics*
- Frédéric Chyzak is on the editorial board of the *Journal of Systems Science and Complexity*.
- Guy Fayolle is associate editor of the journal *Markov Processes and Related Fields (MPRF)*.

Reviewer - reviewing activities

- In 2020, Frédéric Chyzak has been a reviewer for *Journal of Symbolic Computation*, *Transactions of Mathematical Software*, *Journal de l'École polytechnique, Mathématiques*, and for *International Journal of Number Theory*.
- Guy Fayolle has been a reviewer for *Advances in Applied Probability*, *Markov Processes and Related Fields*, *Probability Theory and Related Fields*, *Queueing Systems: Theory and Applications*, *European Journal of Combinatorics*, *Journal of Statistical Physics*, *Physica A*, *Springer Science*.
- In 2020, Alin Bostan has been a reviewer for *Mathematics of Computation*, *Experimental Mathematics*, *Journal of Combinatorial Theory, Series A*, *Séminaire Lotharingien de Combinatoire*, *Journal of Algebraic Combinatorics*, *Bulletin de la Société Mathématique de France*, *Glasgow Mathematical Journal*, *Advances in Applied Mathematics*, *Journal of Applied Analysis*.

7.1.4 Invited talks

- Frédéric Chyzak has been invited to present his work on symbolic integration [5] at the 10th international workshop on Differential Algebra and Related Topics (DART X) (New York, USA) and in the Algorithmic Combinatorics Seminar (Research Institute for Symbolic Computation, Linz).

7.1.5 Scientific expertise

- Guy Fayolle is scientific advisor and associate researcher at the *Robotics Laboratory of Mines ParisTech*.

7.1.6 Research administration

- Guy Fayolle is a member for *Computer System Modeling* of the *International Federation for Information Processing* (IFIP WG 7.3).

7.2 Teaching - Supervision - Juries

7.2.1 Teaching

- **Bachelor**:
  - Alexandre Goyer, *Mathématiques Générales (LSMA100)*, 54h, L1, Université de Versailles Saint-Quentin-en-Yvelines, France.
- **Master**:
  - Alin Bostan, *Algorithmes efficaces en calcul formel*, 36h, M2, MPRI, France.
  - Alin Bostan, *Modern Algorithms for Symbolic Summation and Integration*, 18h, M2, Master d'Informatique Fondamentale de l'ENS de Lyon, France.
  - Frédéric Chyzak, *Algorithmes efficaces en calcul formel*, 36h, M2, MPRI, France. (Also responsible for the course.)
  - Pierre Lairez, *Algorithmique avancée (INF550)*, TD, 18h, M2, École polytechnique, France.
  - Pierre Lairez, *Les bases de la programmation et de l'algorithmique (INF411)*, TD, 40h, M1, École polytechnique, France.

7.2.2 Supervision

- **Bachelor internships**:
  - Frédéric Chyzak co-supervised together with Jérémy Berthomieu (Sorbonne Université) the Bachelor internship of Lucas Morisset on the topic "Fast manipulation of polynomial systems: the F_5 algorithm".
- **Master interships**:
  - Alin Bostan co-supervised together with Xavier Caruso (CNRS, IMB Bordeaux) the Master thesis of Raphaël Pagès on the topic "Computing characteristic polynomials of p-curvatures in average polynomial time". The result was submitted for publication to ISSAC 2021.
  - Frédéric Chyzak co-supervised together with Marc Mezzarobba (CNRS) the Master thesis of Alexandre Goyer on the topic "Symbolic-numeric algorithm for the factorization of differential operators".
- **PhD theses**:
  - Alin Bostan co-supervises together with Xavier Caruso (CNRS, IMB Bordeaux) the PhD thesis of Raphaël Pagès on the topic "Algorithms for factoring linear differential operators in positive characteristic".
  - Alin Bostan co-supervises together with Herwig Hauser (U. Vienna, Austria) the PhD thesis of Sergey Yurkevich on the topic "Integer Sequences arising in Number Theory, Combinatorics and Physics".
  - Frédéric Chyzak co-supervises together with Marc Mezzarobba (CNRS, Lix, Palaiseau) the PhD thesis of Alexandre Goyer on the topic "Symbolic-numeric algorithms in differential algebra".

7.2.3 Juries

- Alin Bostan has served as an examiner in the PhD jury of Youssef Abdelaziz, *Diagonales de fractions rationnelles en physique*, Sorbonne Univ., September 18, 2020.
- Alin Bostan has served as a member of the monitoring PhD committee of Youssef Abdelaziz, Sorbonne Univ.

- Alin Bostan has served as a member of the monitoring PhD committee of Manon Bertin, Univ. Rouen.
- Alin Bostan has served as a member of the monitoring PhD committee of Isabella Panaccione, Ecole polytechnique.
- Frédéric Chyzak has served as a reviewer in the PhD jury of Antonio Jiménez Pastor, *A computable extension for holonomic functions: DD-finite functions*, Johannes Kepler University Linz, December 9, 2020.
- Frédéric Chyzak has served as a reviewer in the mid-PhD examination of Mathilde Chenu, *Study of isogeny-based primitives for post-quantum cryptography*.

8 Scientific production

8.1 Major publications

[1] A. Benoit, A. Bostan and J. Van Der Hoeven. 'Quasi-optimal multiplication of linear differential operators'. In: *FOCS 2012 - IEEE 53rd Annual Symposium on Foundations of Computer Science*. New Brunswick, United States: IEEE, Oct. 2012, pp. 524–530. DOI: 10.1109/FOCS.2012.57. URL: https://hal.archives-ouvertes.fr/hal-00685401.

[2] A. Bostan, M. Bousquet-Mélou and S. Melczer. 'Counting walks with large steps in an orthant'. In: *Journal of the European Mathematical Society* (2020). URL: https://hal.archives-ouvertes.fr/hal-01802706.

[3] A. Bostan, X. Caruso, G. Christol and P. Dumas. 'Fast Coefficient Computation for Algebraic Power Series in Positive Characteristic'. In: *ANTS-XIII - Thirteenth Algorithmic Number Theory Symposium*. Ed. by R. Scheidler and J. Sorenson. Vol. 2. Proceedings of the Thirteenth Algorithmic Number Theory Symposium (ANTS–XIII) 1. Madison, United States: Mathematical Sciences Publishers, July 2018, pp. 119–135. DOI: 10.2140/obs.2019.2-1. URL: https://hal.archives-ouvertes.fr/hal-01816375.

[4] A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, G. Lecerf, B. Salvy and E. Schost. *Algorithmes Efficaces en Calcul Formel*. Voir la page du livre à l'adresse https://hal.archives-ouvertes.fr/AECF/. published by the Authors, 2017. URL: https://hal.inria.fr/hal-01431717.

[5] A. Bostan, F. Chyzak, P. Lairez and B. Salvy. 'Generalized Hermite Reduction, Creative Telescoping and Definite Integration of D-Finite Functions'. In: *ISSAC 2018 - International Symposium on Symbolic and Algebraic Computation*. New York, United States, July 2018, pp. 1–8. DOI: 10.1145/3208976.3208992. URL: https://hal.inria.fr/hal-01788619.

[6] A. Bostan, F. Chyzak, M. Van Hoeij, M. Kauers and L. Pech. 'Hypergeometric Expressions for Generating Functions of Walks with Small Steps in the Quarter Plane'. In: *European Journal of Combinatorics* 61 (2017), pp. 242–275. DOI: 10.1016/j.ejc.2016.10.010. URL: https://hal.inria.fr/hal-01332175.

[7] P. Bürgisser, F. Cucker and P. Lairez. 'Computing the Homology of Basic Semialgebraic Sets in Weak Exponential Time'. In: *Journal of the ACM (JACM)* 66.1 (Dec. 2018), pp. 1–30. DOI: 10.1145/3275242. URL: https://hal.archives-ouvertes.fr/hal-01545657.

[8] F. Chyzak, T. Dreyfus, P. Dumas and M. Mezzarobba. 'Computing solutions of linear Mahler equations'. In: *Mathematics of Computation* 87 (July 2018), pp. 2977–3021. DOI: 10.1090/mcom/3359. URL: https://hal.inria.fr/hal-01418653.

[9] F. Chyzak, A. Mahboubi, T. Sibut-Pinote and E. Tassi. 'A Computer-Algebra-Based Formal Proof of the Irrationality of $\zeta(3)$'. In: *ITP - 5th International Conference on Interactive Theorem Proving*. Vienna, Austria, 2014. URL: https://hal.inria.fr/hal-00984057.

[10] P. Lairez. 'Computing periods of rational integrals'. In: *Mathematics of Computation* 85 (Nov. 2016), pp. 1719–1752. DOI: 10.1090/mcom/3054. URL: https://hal.inria.fr/hal-00981114.

8.2 Publications of the year

International journals

[11] V. Andrejić, A. Bostan and M. Tatarevic. 'Improved algorithms for left factorial residues'. In: *Information Processing Letters*. 167th ser. (2021). DOI: 10.1016/j.ipl.2020.106078. URL: https://hal.archives-ouvertes.fr/hal-02411741.

[12] A. Bostan, M. Bousquet-Mélou and S. Melczer. 'Counting walks with large steps in an orthant'. In: *Journal of the European Mathematical Society* (2020). URL: https://hal.archives-ouvertes.fr/hal-01802706.

[13] A. Bostan, F. Chamizo and M. Sundqvist. 'On an Integral Identity'. In: *American Mathematical Monthly* (2020). URL: https://hal.inria.fr/hal-03084594.

[14] A. Bostan, F. Chyzak, A. Jiménez-Pastor and P. Lairez. 'The Sage Package comb_walks for Walks in the Quarter Plane'. In: *ACM Communications in Computer Algebra* 54.212 (2020), p. 9. DOI: 10.1145/3427218.3427220. URL: https://hal.inria.fr/hal-02902709.

[15] A. Bostan, A. Elvey-Price, A. J. Guttmann and J.-M. Maillard. 'Stieltjes moment sequences for pattern-avoiding permutations'. In: *The Electronic Journal of Combinatorics* (2020), p. 59. DOI: 10.37236/xxxx. URL: https://hal.archives-ouvertes.fr/hal-02425917.

[16] A. Bostan and A. Jiménez-Pastor. 'On the exponential generating function of labelled trees'. In: *Comptes rendus de l'Académie des sciences. Série I, Mathématique* (2020). URL: https://hal.inria.fr/hal-03084481.

[17] A. Bostan, T. Krick, A. Szanto and M. Valdettaro. 'Subresultants of $(x-\alpha)^m$ and $(x-\beta)^n$, Jacobi polynomials and complexity'. In: *Journal of Symbolic Computation* 101 (2020), pp. 330–351. DOI: 10.1016/j.jsc.2019.10.003. URL: https://hal.archives-ouvertes.fr/hal-01966640.

[18] A. Bostan, T. Rivoal and B. Salvy. 'Explicit degree bounds for right factors of linear differential operators'. In: *Bulletin of the London Mathematical Society* 53.1 (1st Feb. 2021), pp. 53–62. DOI: 10.1112/blms.12396. URL: https://hal.archives-ouvertes.fr/hal-02154679.

[19] C. Boutillier, K. Raschel and A. Bostan. 'Martin boundary of killed random walks on isoradial graphs'. In: *Potential Analysis* (2021). URL: https://hal.archives-ouvertes.fr/hal-02422417.

[20] F. Chapoton and A. Bostan. 'A note on gamma triangles and local gamma vectors'. In: *Annales de la Faculté des Sciences de Toulouse. Mathématiques*. Série 6, Tome 29 4 (9th Dec. 2020), pp. 907–925. URL: https://hal.archives-ouvertes.fr/hal-01866199.

[21] G. Fayolle and R. Iasnogorodski. 'Conditions for some non stationary random walks in the quarter plane to be singular or of genus 0'. In: *Markov Processes And Related Fields* (Mar. 2021). URL: https://hal.inria.fr/hal-03008556.

International peer-reviewed conferences

[22] A. Bostan. 'Computing the N-th Term of a q-Holonomic Sequence'. In: *ISSAC'20: Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation*. ISSAC 2020 - 45th International Symposium on Symbolic and Algebraic Computation. Kalamata, Greece, 20th July 2020, p. 8. DOI: 10.1145/3373207.3404060. URL: https://hal.inria.fr/hal-02882885.

[23] A. Bostan, A. Carayol, F. Koechlin and C. Nicaud. 'Weakly-unambiguous Parikh automata and their link to holonomic series'. In: ICALP 2020 - 47th International Colloquium on Automata, Languages and Programming. Saarbrücken, Germany: https://icalp2020.saarland-informatics-campus.de, 8th July 2020, p. 16. URL: https://hal.inria.fr/hal-03084639.

[24] A. Bostan and R. Mori. 'A Simple and Fast Algorithm for Computing the N-th Term of a Linearly Recurrent Sequence'. In: SOSA'21 (SIAM Symposium on Simplicity in Algorithms). Alexandria, United States: https://www.siam.org/conferences/cm/conference/sosa21, 11th Jan. 2021. URL: https://hal.inria.fr/hal-02917827.

[25] F. Chyzak and P. Dumas. 'A Gröbner-Basis Theory for Divide-and-Conquer Recurrences'. In: *ISSAC'20: Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation*. ISSAC - 2020 - 45th International Symposium on Symbolic and Algebraic Computation. Kalamata, Greece, 20th July 2020. DOI: 10.1145/3373207.3404055. URL: https://hal.inria.fr/hal-02885579.

[26] S. Yurkevich. 'Diagonal Representation of Algebraic Power Series: A Glimpse Behind the Scenes'. In: *Transient Transcendence in Transylvania*. This work is based on the author's master's thesis (U. of Vienna, 2020), supervised by H. Hauser. Brașov, Romania, May 2019. URL: https://hal.archives-ouvertes.fr/hal-03150958.

Reports & preprints

[27] A. Bostan, L. Di Vizio and K. Raschel. *Differential transcendence of Bell numbers and relatives: a Galois theoretic approach*. 30th Dec. 2020. URL: https://hal.archives-ouvertes.fr/hal-03091272.

[28] A. Bostan, M. Kauers and T. Verron. *The generating function of Kreweras walks with interacting boundaries is not algebraic*. 2020. URL: https://hal.inria.fr/hal-03084659.

[29] A. Bostan and S. Yurkevich. *Fast Computation of the N-th Term of a q-Holonomic Sequence and Applications*. 2020. URL: https://hal.inria.fr/hal-03084680.

[30] A. Bostan and S. Yurkevich. *On a Class of Hypergeometric Diagonals*. 2020. URL: https://hal.inria.fr/hal-03084672.

[31] P. Bürgisser, F. Cucker and P. Lairez. *Rigid continuation paths II. Structured polynomial systems*. 21st Oct. 2020. URL: https://hal.archives-ouvertes.fr/hal-02974062.

[32] F. Chyzak and K. Yeats. *Bijections between Łukasiewicz walks and generalized tandem walks*. 17th Feb. 2020. URL: https://hal.inria.fr/hal-01891792.

[33] G. Fayolle. *A note on the connection between product-form Jackson networks and counting lattice walks in the quarter plane*. 6th Jan. 2020. URL: https://hal.inria.fr/hal-02415746.

[34] G. Gonthier and L. Lamport. *Recursive Operator Definitions*. Inria Saclay Ile de France, May 2020, p. 17. URL: https://hal.inria.fr/hal-02598330.

[35] P. Lairez and E. Can Sertöz. *Separation of periods of quartic surfaces*. 1st Dec. 2020. URL: https://hal.archives-ouvertes.fr/hal-03022612.

8.3 Cited publications

[36] Y. Abdelaziz, C. Koutschan and J.-M. Maillard. 'On Christol's conjecture'. In: *J. Phys. A* 53.20 (2020), 205201, 16 pages. DOI: 10.1088/1751-8121/ab82dc. URL: https://doi.org/10.1088/1751-8121/ab82dc.

[37] M. Abramowitz and I. A. Stegun, eds. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Reprint of the 1972 edition. New York: Dover, 1992, pp. xiv+1046.

[38] M. Armand, B. Grégoire, A. Spiwack and L. Théry. 'Extending Coq with Imperative Features and its Application to SAT Verication'. In: *Interactive Theorem Proving, international Conference, ITP 2010, Edinburgh, Scotland, July 11–14, 2010, Proceedings*. Lecture Notes in Computer Science. Springer, 2010.

[39] B. Beckermann and G. Labahn. 'A uniform approach for the fast computation of matrix-type Padé approximants'. In: *SIAM J. Matrix Anal. Appl.* 15.3 (1994), pp. 804–823.

[40] A. Benoit, F. Chyzak, A. Darrasse, S. Gerhold, M. Mezzarobba and B. Salvy. 'The Dynamic Dictionary of Mathematical Functions (DDMF)'. In: *The Third International Congress on Mathematical Software (ICMS 2010)*. Ed. by K. Fukuda, J. van der Hoeven, M. Joswig and N. Takayama. Vol. 6327. Lecture Notes in Computer Science. 2010, pp. 35–41.

[41] M. Boespflug, M. Dénès and B. Grégoire. 'Full reduction at full throttle'. In: *First International Conference on Certified Programs and Proofs, Taiwan, December 7–9*. Lecture Notes in Computer Science. Springer, 2011.

[42] S. Boldo, C. Lelay and G. Melquiond. 'Improving Real Analysis in Coq: A User-Friendly Approach to Integrals and Derivatives'. In: *Certified Programs and Proofs*. Ed. by C. Hawblitzel and D. Miller. Vol. 7679. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 289–304. URL: http://dx.doi.org/10.1007/978-3-642-35308-6_22.

[43] S. Boldo and G. Melquiond. 'Flocq: A Unified Library for Proving Floating-point Algorithms in Coq'. In: *Proceedings of the 20th IEEE Symposium on Computer Arithmetic*. Tübingen, Germany, July 2011, pp. 243–252.

[44] A. Bostan. 'Algorithmes rapides pour les polynômes, séries formelles et matrices'. In: *Actes des Journées Nationales de Calcul Formel*. Les cours du CIRM, tome 1, numéro 2. Luminy, France, 2010, pp. 75–262. URL: http://ccirm.cedram.org:80/ccirm-bin/fitem?id=CCIRM_2010__1_2_75_0.

[45] A. Bostan, S. Boukraa, S. Hassani, J.-M. Maillard, J.-A. Weil and N. Zenine. 'Globally nilpotent differential operators and the square Ising model'. In: *J. Phys. A: Math. Theor.* 42.12 (2009), p. 50. URL: http://dx.doi.org/10.1088/1751-8113/42/12/125206.

[46] A. Bostan, S. Chen, F. Chyzak and Z. Li. 'Complexity of creative telescoping for bivariate rational functions'. In: *ISSAC'10: Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*. New York, NY, USA: ACM, 2010, pp. 203–210. URL: http://doi.acm.org/10.1145/1837934.1837975.

[47] A. Bostan, F. Chyzak, M. van Hoeij and L. Pech. 'Explicit formula for the generating series of diagonal 3D rook paths'. In: *Sém. Loth. Comb.* B66a (2011), p. 27. URL: http://www.emis.de/journals/SLC/wpapers/s66bochhope.html.

[48] A. Bostan, F. Chyzak, G. Lecerf, B. Salvy and É. Schost. 'Differential equations for algebraic functions'. In: *ISSAC'07: Proceedings of the 2007 international symposium on Symbolic and algebraic computation*. Ed. by C. W. Brown. ACM Press, 2007, pp. 25–32. URL: http://dx.doi.org/10.1145/1277548.1277553.

[49] A. Bostan and M. Kauers. 'The complete generating function for Gessel walks is algebraic'. In: *Proceedings of the American Mathematical Society* 138.9 (2010). With an appendix by Mark van Hoeij, pp. 3063–3078.

[50] F. Chyzak. 'An extension of Zeilberger's fast algorithm to general holonomic functions'. In: *Discrete Math.* 217.1-3 (2000). Formal power series and algebraic combinatorics (Vienna, 1997), pp. 115–134.

[51] F. Chyzak, M. Kauers and B. Salvy. 'A Non-Holonomic Systems Approach to Special Function Identities'. In: *ISSAC'09: Proceedings of the Twenty-Second International Symposium on Symbolic and Algebraic Computation*. Ed. by J. May. 2009, pp. 111–118. URL: http://dx.doi.org/10.1145/1576702.1576720.

[52] F. Chyzak and B. Salvy. 'Non-commutative elimination in Ore algebras proves multivariate identities'. In: *J. Symbolic Comput.* 26.2 (1998), pp. 187–227.

[53] T. Coquand and G. P. Huet. 'The Calculus of Constructions'. In: *Inf. Comput.* 76.2/3 (1988), pp. 95–120. URL: http://dx.doi.org/10.1016/0890-5401(88)90005-3.

[54] T. Coquand and C. Paulin-Mohring. 'Inductively defined types'. In: *Proceedings of Colog'88*. Ed. by P. Martin-Löf and G. Mints. Vol. 417. Lecture Notes in Computer Science. Springer-Verlag, 1990.

[55] D. Delahaye and M. Mayero. 'Dealing with algebraic expressions over a field in Coq using Maple'. In: *J. Symbolic Comput.* 39.5 (2005). Special issue on the integration of automated reasoning and computer algebra systems, pp. 569–592. URL: http://dx.doi.org/10.1016/j.jsc.2004.12.004.

[56] G. Fayolle, R. Iasnogorodski and V. Malyshev. *Random walks in the quarter plane*. Springer International Publishing, 2017. DOI: 10.1007/978-3-319-50930-3.

[57] F. Garillot, G. Gonthier, A. Mahboubi and L. Rideau. 'Packaging Mathematical Structures'. In: *Theorem Proving in Higher-Order Logics*. Ed. by S. Berghofer, T. Nipkow, C. Urban and M. Wenzel. Vol. 5674. Lecture Notes in Computer Science. Springer, 2009, pp. 327–342.

[58] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. 2nd. New York: Cambridge University Press, 2003, pp. xiv+785.

[59] G. Gonthier. 'Formal proofs—the four-colour theorem'. In: *Notices of the AMS* 55.11 (2008), pp. 1382–1393.

[60] G. Gonthier and A. Mahboubi. 'An introduction to small scale reflection in Coq'. In: *Journal of Formalized Reasoning* 3.2 (2010), pp. 95–152.

[61] G. Gonthier, A. Mahboubi and E. Tassi. *A Small Scale Reflection Extension for the Coq system*. Anglais. Rapport de recherche RR-6455. INRIA, 2008. URL: http://hal.inria.fr/inria-00258384.

[62] G. Gonthier and E. Tassi. 'A language of patterns for subterm selection'. In: *ITP*. Vol. 7406. LNCS. 2012, pp. 361–376.

[63] B. Grégoire and A. Mahboubi. 'Proving Equalities in a Commutative Ring Done Right in Coq'. In: *Theorem Proving in Higher Order Logics, 18th International Conference, TPHOLs 2005, Oxford, UK, August 22-25, 2005, Proceedings*. Vol. 3603. Lecture Notes in Computer Science. Springer, 2005, pp. 98–113.

[64] T. Hales. 'Formal proof'. In: *Notices of the AMS* 55.11 (2008), pp. 1370–1380.

[65] J. Harrison. 'A HOL Theory of Euclidean space'. In: *Theorem Proving in Higher Order Logics, 18th International Conference, TPHOLs 2005*. Ed. by J. Hurd and T. Melham. Vol. 3603. Lecture Notes in Computer Science. Oxford, UK: Springer-Verlag, 2005, pp. 114–129.

[66] J. Harrison. 'A Machine-Checked Theory of Floating Point Arithmetic'. In: *Theorem Proving in Higher Order Logics: 12th International Conference, TPHOLs'99*. Ed. by Y. Bertot, G. Dowek, A. Hirschowitz, C. Paulin and L. Théry. Vol. 1690. Lecture Notes in Computer Science. Nice, France: Springer-Verlag, 1999, pp. 113–130.

[67] J. Harrison. 'Formalizing an analytic proof of the prime number theorem'. In: *Journal of Automated Reasoning* 43 (2009). Dedicated to Mike Gordon on the occasion of his 60th birthday, pp. 243–261.

[68] J. Harrison. *Theorem proving with the real numbers*. CPHC/BCS distinguished dissertations. Springer, 1998.

[69] J. Harrison and L. Théry. 'A Skeptic's Approach to Combining HOL and Maple'. In: *J. Autom. Reason.* 21.3 (Dec. 1998), pp. 279–294. URL: http://dx.doi.org/10.1023/A:1006023127567.

[70] F. Johansson. *Another Mathematica bug*. Article on personal blog. 2009. URL: http://fredrik-j.blogspot.fr/2009/07/another-mathematica-bug.html.

[71] C. Koutschan. 'A fast approach to creative telescoping'. In: *Math. Comput. Sci.* 4.2-3 (2010), pp. 259–266. DOI: 10.1007/s11786-010-0055-0. URL: http://dx.doi.org/10.1007/s11786-010-0055-0.

[72] A. Mahboubi. 'Implementing the cylindrical algebraic decomposition within the Coq system'. In: *Mathematical Structures in Computer Science* 17.1 (2007), pp. 99–127.

[73] *Computer Algebra Errors*. Article in mathematics blog *MathOverflow*. URL: http://mathoverflow.net/questions/11517/computer-algebra-errors.

[74] R. Matuszewski and P. Rudnicki. 'Mizar: the first 30 years'. In: *Mechanized Mathematics and Its Applications* 4 (2005).

[75] M. Mayero. 'Problèmes critiques et preuves formelles'. Habilitation à Diriger des Recherches. Université Paris 13, 2012.

[76] M. Mezzarobba. 'NumGfun: a package for numerical and analytic computation and D-finite functions'. In: *ISSAC 2010—Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*. New York: ACM, 2010, pp. 139–146. DOI: 10.1145/1837934.1837965. URL: http://dx.doi.org/10.1145/1837934.1837965.

[77] F. W. J. Olver, D. W. Lozier, R. F. Boisvert and C. W. Clark, eds. *NIST Handbook of mathematical functions*. Cambridge University Press, 2010.

[78] P. Paule and M. Schorn. 'A Mathematica version of Zeilberger's algorithm for proving binomial coefficient identities'. In: *J. Symbolic Comput.* 20.5-6 (1995). Symbolic computation in combinatorics Δ_1 (Ithaca, NY, 1993), pp. 673–698. DOI: 10.1006/jsco.1995.1071. URL: http://dx.doi.org/10.1006/jsco.1995.1071.

[79] B. Petersen. *Maple*. Personal web site.

[80] P. Rudnicki and A. Trybulec. 'On the Integrity of a Repository of Formalized Mathematics'. In: *Proceedings of the Second International Conference on Mathematical Knowledge Management*. MKM '03. London, UK: Springer-Verlag, 2003, pp. 162–174. URL: http://dl.acm.org/citation.cfm?id=648071.748518.

[81] B. Salvy and P. Zimmermann. 'Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable'. In: *ACM Trans. Math. Software* 20.2 (1994), pp. 163–177.

[82] N. J. A. Sloane and S. Plouffe. *The Encyclopedia of Integer Sequences*. Academic Press, San Diego, 1995.

[83] The Coq Development Team. *The Coq Proof Assistant: Reference Manual*. URL: http://coq.inria.fr/doc/.

[84] The Mathematical Component Team. *A Formalization of the Odd Order Theorem using the Coq proof assistant*. 2012. URL: http://www.msr-inria.fr/projects/mathematical-components/.

[85] L. Théry. 'A Machine-Checked Implementation of Buchberger's Algorithm'. In: *J. Autom. Reasoning* 26.2 (2001), pp. 107–137. URL: http://dx.doi.org/10.1023/A:1026518331905.

[86] K. Wegschaider. 'Computer generated proofs of binomial multi-sum identities'. Diplomarbeit. RISC, J. Kepler University, May 1997, p. 99.

[87] S. Wolfram. *Mathematica: A system for doing mathematics by computer (2nd ed.)* Addison-Wesley, 1992.

[88] D. Zeilberger. 'A holonomic systems approach to special functions identities'. In: *J. Comput. Appl. Math.* 32.3 (1990), pp. 321–368.

[89] D. Zeilberger. *Opinion 94: The Human Obsession With "Formal Proofs" is a Waste of the Computer's Time, and, Even More Regretfully, of Humans' Time*. 2009. URL: http://www.math.rutgers.edu/~zeilberg/Opinion94.html.

[90] D. Zeilberger. 'The method of creative telescoping'. In: *J. Symbolic Comput.* 11.3 (1991), pp. 195–204.