Do Creative-Telescoping Algorithms Provide Complete Proofs? A Formal Study of Apéry's Theorem*

Frédéric Chyzak
Inria (France)
frederic.chyzak@inria.fr

Assia Mahboubi
Inria (France)
assia.mahboubi@inria.fr

Thomas Sibut-Pinote
ENS Lyon (France)
thomas.sibutpinote@ens-lyon.fr

ABSTRACT

We report on the formal verification of an irrationality proof of $\zeta(3)$, the evaluation of the Riemann zeta function. This verification uses the Coq proof assistant in conjunction with algorithmic calculations in Maple. This experience illustrates the limits of the common belief that creative-telescoping algorithms can discover recurrences for holonomic sequences that are easy to check a posteriori. We discuss this observation and describe the protocol we devised in order to produce *complete* formal proofs of the recurrences.

Categories and Subject Descriptors:
G.2.1 [**Mathematics of Computing**]: Discrete Mathematics — *Recurrences and difference equations*

General Terms: Theory, Verification.

Keywords: creative telescoping, formalization, irrationality proof.

1. INTRODUCTION

Computer algebra is primarily about computing, whether it be simplifying an expression, solving (like a linear or polynomial system, an ODE or a recurrence), approximating (e.g., a function by a series), or changing representations (like changing the recursive representation of multivariate polynomials or obtaining a better basis for a vector space, an ideal, or another algebraic structure). But, always, computations claim the status of proofs, as each one states some kind of an identity. These identities are justified, in principle, by theorems *on paper*, supposed to establish the correctness of the algorithms used, and, of course, by the act of faith that the implementation matches the intention of the algorithm.

Additionally, specifically when computer algebra is used in applications to obtain a new proof of a mathematical fact, or the first proof of a conjecture [14, 16, 21, 22, 23, 24, 26], calculations are augmented with more paper proofs that interpret the successive results and lead to the final mathematical statement. This particular choice of examples illustrates the success of algorithms computing properties (identities, asymptotics, ...) on a large class of *sequences*. The confidence in the calculations performed by these algorithms about sequences is increased by the common sense conveyed by the literature that these proofs on sequences can be justified a posteriori, that is, after calling the computer-algebra algorithms.

Formalizing mathematics consists in providing a precise and unambiguous representation of mathematical objects, of their properties, and of the proofs thereof, in the codified language of logic. Candidate proofs of mathematical statements become this way amenable to mechanical checking, by a single program that can be trusted because it is small and simple. Interactive proof assistants are pieces of mathematical software that aim at easing the tasks of human formalization and machine checking. In addition, the activity of formalizing mathematics often requires polishing the definitions of the mathematical objects at stake and scrutinizing the associated patterns of reasoning. Indeed, the proof checker of a proof assistant is, on purpose, insensitive to implicit proof steps or analogies without which it is not possible to communicate mathematics in a way intelligible to a human reader.

Turning a computer-algebra proof into a proof checked by a proof assistant requires several ingredients. If part of the proof consists in calculations that are easy to check a posteriori, it is often relevant to take benefit of this situation and to privilege a skeptical approach to formal certification [20]: a computer-algebra program can be used as an *external* oracle in order to produce conjectures that are verified a posteriori by a calculation performed *inside the logic*. This verification typically consists in computing and comparing normal forms of algebraic objects (e.g., arithmetical expressions, rational fractions). Moreover, the paper part of the proof is replaced by a machine-checked formal proof, gradually elaborated by its author through an interaction with the proof assistant. Note that a proof assistant only knows about the syntax of logical objects and does not assign any semantics to expressions, as opposed for instance to the built-in notion of arithmetic expressions typically featured by a computer-algebra system. Hence a formalized proof starts by defining inside the logic the various mathematical objects (e.g., sequences, binomials, polynomials) the theorem is about. In particular, if computer algebra calculates with some algebraic abstraction of a concrete, analytic object, the formalization should make explicit a correct interpretation of the computational results on the original object.

*Supported in part by the Microsoft Research – Inria Joint Centre.

ISSAC'14, July 21–25, 2014, Kobe, Japan.

We have completed a formal proof of irrationality of $\zeta(3)$ by using the Coq proof assistant [34] in cooperation with the computer-algebra system Maple. In particular, this formalization includes a formal a posteriori verification of the computer-algebra calculations. Prior to this experience, we shared the common belief that such a formal a posteriori verification, tedious and error-prone if performed by hand, could be automated easily. However, the present work revealed the limits of the common belief. Actually the algorithms run by the computer-algebra session are specified in the litterature for objects that do not match the nature of concrete sequences involved in the irrationality proof. It happens that the ease of a posteriori checking is compromised by this discrepancy.

In the present article, we focus on the part of our formalized irrationality proof of $\zeta(3)$ devoted to the validation of computer-algebra calculations. (Other aspects not pertaining to computer algebra will be described in the related article [10].) Our first contribution is taking a critical look at symbolic-summation algorithms developed since the 1990s by the *approach of creative telescoping*. Next, Lemma 1 precisely justifies how creative telescoping turns specific recurrences for a given summand into a recurrence for its definite sum. This turns into an explicit statement what is found only as a method worked out on examples in the literature. Last, we devised a protocol to formally validate the recurrences obtained by computer algebra when combining ∂-finite sequences by their closure operations. We identified procedures for these closures, which all base on rewriting modulo *recurrences with provisos*. Unfortunately, we could not turn these procedures into complete algorithms yet.

The proof by Apéry on which we base our work is described in Section 2, together with alternatives. Section 3 provides a critical view on the commonly accepted approach of *creative telescoping* for proving combinatorial sums and related identities. Creative telescoping is often described in its simplest form in the literature. In the present paper, we address the case of creative telescoping for sums with varying bounds in presence of singularities by Lemma 1 in Section 4. Our proof of irrationality bases crucially on successively obtaining recurrences for each of the sequences in (6) below. The formalization of how to obtain these recurrences is described in Section 5. Finally, we give conclusions and perspectives in Section 6.

Our twin paper [10] as well as our Maple and Coq scripts will be found at `http://specfun.inria.fr/zeta-of-3/`.

Notation

In this work, we consider sequences of one or two integer indices with values in a field K, that is, functions u from either $K^{\mathbb{Z}}$ or $u \in K^{\mathbb{Z}^2}$ with values at n, resp. (n,k), denoted by u_n, resp. $u_{n,k}$. If u is a bivariate sequence and $j \in \mathbb{Z}$ is a fixed integer, $u_{j,\_}$, resp. $u_{\_,j}$, denotes the univariate sequence obtained by specializing the first, resp. the second, argument of u to j. In addition, an *operator* on sequences is a (total) map from K^s to itself ($s = 1$ or $s = 2$).

2. MATHEMATICAL AND ALGORITHMICAL CONTEXT

While the evaluations of the Riemann zeta function at positive even integers are known to lie in $\mathbb{Q}(\pi)$ and those at nonpositive integers in $\mathbb{Q}$, not much is known about the rationality or irrationality of its evaluations at positive odd integers. It was therefore a great breakthrough in 1978 when Apéry proved that $\zeta(3)$ is irrational [3, 4]. Twenty years later, Rivoal proved the existence of infinitely many irrational values at odd integers [27], without being able to name any but $\zeta(3)$, and that one of the numbers $\zeta(5)$, $\zeta(7)$, $\ldots$, $\zeta(21)$ is irrational [28]; this interval was then narrowed down to $\zeta(5)$, $\ldots$, $\zeta(11)$ by Zudilin [45].

2.1 Apéry's and Beukers' Proofs

Beside Apéry's rather terse original presentation, an explanatory one was proposed in [35]. For his proof, Apéry introduced two sequences of rational numbers $(a_n)_{n\in\mathbb{N}}$ and $(b_n)_{n\in\mathbb{N}}$ such that $\zeta(3)$ is the limit of the quotients b_n/a_n. The proof is completed by a classical number-theoretic argument that "too many" quotients are "too close" to $\zeta(3)$. To obtain that b_n/a_n is "close enough" for the proof to work, the a_n and b_n are obtained by Legendre transformation, starting from the initial approximation

$$u_{n,k} = \sum_{m=1}^{n} \frac{1}{m^3} + \sum_{m=1}^{k} \frac{(-1)^{m+1}}{2m^3\binom{n}{m}\binom{n+m}{m}},$$

which tends to $\zeta(3)$ when n goes to infinity. This leads to

$$a_n = \sum_{k=1}^{n} c_{n,k}, \quad b_n = \sum_{k=1}^{n} c_{n,k}u_{n,k}, \quad \text{for } c_{n,k} = \binom{n}{k}^2\binom{n+k}{k}^2. \tag{1}$$

Introduce the lcm ℓ_n of the integers $1,\ldots,n$. Apéry's proof can be organized in four main parts: *(i)* using elementary number theory in order to establish that $2\ell_n^3 b_n$ is an integer; *(ii)* proving that $(a_n)_{n\in\mathbb{N}}$ and $(b_n)_{n\in\mathbb{N}}$ both satisfy the *same* second-order recurrence

$$(n+1)^3 y_{n+1} - (34n^3+51n^2+27n+5)y_n + n^3 y_{n-1} = 0; \tag{2}$$

(iii) deriving useful consequences of this recurrence, namely the positivity of $\delta_n = a_n\zeta(3) - b_n$ and that the sequence $(\delta_n)_{n\in\mathbb{N}}$ is asymptotically infinitesimally small; *(iv)* assuming that $\zeta(3)$ is rational and combining the previous facts with an estimation of the asymptotic of $(\ell_n)_{n\in\mathbb{N}}$ to conclude a contradiction.

In the litterature, the asymptotic study to get the bound $\ell_n = e^{n\,(1+o(1))}$ bases on the distribution of the prime numbers. Other more elementary bounds that are tight enough for our purpose exist. Notably, independent elementary proofs of a bound 3^n are given by Hanson [19] and Feng [15].

An alternative, shorter and more elegant proof was proposed by Beukers [6], who interpreted Apéry's approximations by integrals. From the definition of Legendre polynomials $L_n(x)$ as an nth derivative and the fact that

$$\int_0^1\int_0^1\int_0^1 \frac{L_n(x)\,L_n(y)}{1-u\,(1-xy)}dx\,dy\,du = \big(A_n + B_n\,\zeta(3)\big)\,\ell_n^3$$

for integers A_n and B_n, Beukers derived by integration by parts that the integral above is nonzero and asymptotically small. The irrationality of $\zeta(3)$ follows by the same final arguments as in Apéry's proof.

2.2 Recurrences as a Data Structure

In the 1990s, combinatorialists and computer-algebraists got interested in designing algorithms to "compute" expressions like $u_{n,k}$, a_n, and b_n above: with procedures like Zeilberger's algorithm [43] and its extension [8] for single sums,

or [37] for multiple sums, it became possible to determine recurrences satisfied by those sequences algorithmically. Even when no closed form can be obtained from these outputs, much information can be extracted. This is so in Apéry's proof, which derives the asymptotic bound on and the positivity of δ_n from the second-order recurrence (2) satisfied simultaneously by (a_n) and (b_n).

In fact, this approach, largely initiated by Zeilberger [42], promoted (linear) recurrences as the right representation of a large class of sequences closed under many operations, nowadays known as ∂-finite sequences [11]: a sequence is described by a linear recurrence or set of linear recurrences whose solution set is a finite-dimensional vector space, decorated with a sufficient finite number of initial conditions. Effective procedures for closures in the univariate case are folklore and can be found implicitly in [33], and have been extended to the multivariate case in the 1990s [42, 11]. For simple operations (addition, product, shift, and similar combinations), algorithms reduce to linear algebra for obtaining linear dependencies between shifts of the composite sequences to be described. Additionally, Zeilberger borrowed from D-module theory the setup of holonomic systems to guarantee the existence of linear recurrences for the definite sum of a sequence given by its set of linear recurrences in one more index. Zeilberger designed algorithms specific to the case of hypergeometric sequences [41, 38], which were later extended to larger classes of inputs [8, 9].

Zeilberger's approach to summation bases on an operation named *creative telescoping*, a term coined by van der Poorten [35]. Given recurrences for a hypergeometric summand $u_{n,k}$ to be summed for k between integers α and β (independent of n), thus considering $U_n = \sum_{k=\alpha}^{\beta} u_{n,k}$, it consists in first obtaining a relation of a specific shape:

$$p_r(n)\,u_{n+r,k} + \cdots + p_0(n)\,u_{n,k} = q(n,k+1)\,u_{n,k+1} - q(n,k)\,u_{n,k}, \quad (3)$$

for some integer r, polynomials p_i independent from k, and a bivariate rational function q. The motivation is that summing over k (provided this makes sense) delivers

$$p_r(n)\,U_{n+r} + \cdots + p_0(n)\,U_n = q(n,\beta+1)\,u_{n,\beta+1} - q(n,\alpha)\,u_{n,\alpha}, \quad (4)$$

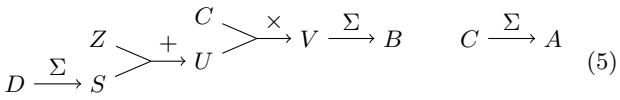
where the right-hand side has been obtained by a telescoping sum, giving its name to the method. In nice cases, this right-hand side evaluates to 0, which yields a linear homogeneous recurrence for U; in other cases, the evaluations $u_{n,a}$ and $u_{n,b+1}$ satisfy recurrences themselves, which can be recombined to cancel the right-hand side, and also provide a linear homogeneous recurrence for U by composition.

Beside algorithms, the computer-algebra community came up with implementations to manipulate ∂-finite sequences and recurrences they satisfy, notably, the Maple package Gfun [30] for univariate sequences (among other things) and its multivariate counterpart Mgfun by Chyzak, both distributed as parts of the Algolib library [1]. Based on them, Salvy wrote a Maple worksheet [29] that completes a proof of irrationality of $\zeta(3)$ by Apéry's approach, letting Maple perform all calculations needed, and interlacing them with human-written logical steps as comments in the session.

2.3 The Crucial Recurrence in Apéry's Proof

Salvy's worksheet and our Coq formalization share the structure described in Section 2.1. For the crucial step of deriving the recurrence for $(a_n)_{n\in\mathbb{N}}$ and $(b_n)_{n\in\mathbb{N}}$, they are guided by calculations performed by a Maple script, appealing to the Algolib library. But they differ in that the computer-algebra worksheet views Maple calculations as proof steps, while our Coq proof follows the skeptical approach [20] already discussed in Section 1. Additionally, in our Coq proof the hand-written parts of the Maple worksheet are replaced by machine-checked formal proofs.

The calculations leading to recurrence (2) can be viewed as the program



(5)

where each of the node labels A, B, C, D, S, U, V, and Z has to be understood as recurrences to be computed (together with sufficiently many initial conditions) for the corresponding sequence in the list:

$$c_{n,k} = \binom{n}{k}^2 \binom{n+k}{k}^2, \quad d_{n,m} = \frac{(-1)^{m+1}}{2m^3\binom{n}{m}\binom{n+m}{m}},$$
$$z_n = \sum_{m=1}^{n} \frac{1}{m^3}, \quad a_n = \sum_{k=1}^{n} c_{n,k}, \quad s_{n,k} = \sum_{m=1}^{k} d_{n,m}, \quad (6)$$
$$u_{n,k} = z_n + s_{n,k}, \quad v_{n,k} = c_{n,k}u_{n,k}, \quad b_n = \sum_{k=1}^{n} v_{n,k}.$$

The program (5) precisely follows the syntax trees defining the sequences a and b, so that the edges in the program are labelled with the closure operation performed by the program. More explicitly, the sequences z, c, and d are defined directly by recurrences that are easy to deduce from their closed forms (see, e.g., (9) below), while other sequences are derived through closure operations: summation for a, s, and b; addition for u; product for v. In fact, the program above is a minor reordering of Salvy's presentation. But with respect to proof, this is not quite innocent, as it permitted a uniform procedural treatment of all sums.

As it turns out, the algorithms mentioned in Section 2.2 are precisely specified for the tasks in the program (5), except that they generally do not maintain initial conditions. Thus, the algorithm employed at a node returns recurrences that hold in fact not just for the specific sequences satisfying the input systems in the program, but for any choice of sequence solutions of the prescribed input systems.

3. "PROOFS" ON ∂-FINITE SEQUENCES

In the early 1990s, Zeilberger popularized the idea that a large class of mathematical identities, whether combinatorial or about special functions, had become *routinely verifiable* on a computer, using computer-algebra algorithms. He presented his views in sometimes disputable pamphlets [40, 37, 44] as well as in theoretical papers [41, 42, 38]. This was accompanied by many articles proving sample identities by the method and was followed by the book [25]. We suggest the reader look up the many computer-aided proofs of identities "co-authored" by (the computer) Shalosh B. Ekhad and Zeilberger, to be found at http://www.math.rutgers.edu/~zeilberg/pj.html.

The treatment of additions and products of sequences, as suggested in [42, Section 4.1 and 4.2] and continued in [11,

Lemmas 2.1 and 2.2], as well as the treatment of summation in [43, 8], rely on the assertion that verifying an identity

$$\sum_{(i,j)\in S} c_{i,j}(n,k) f_{n+i,k+j} = 0, \tag{7}$$

on a sequence $(f_{n,k})_{(n,k)\in\mathbb{Z}^2}$ for polynomials $c_{i,j}$ and a finite set of shifts $S \subset \mathbb{N}^2$, reduces to simplifying according to rules of the form

$$f_{n+I,k+J} = \sum_{(i,j)\in U} c'_{i,j}(n,k) f_{n+i,k+j}, \tag{8}$$

for a common set $U \subset \mathbb{N}^2$, a finite set of pairs (I, J), and *rational* functions $c'_{i,j}$. As was developed in [11], the proper set of rules can be described in terms of *Gröbner bases*: when the family of the relations (8) is a Gröbner basis, any path of reduction of the left-hand side of (7) modulo the (8) terminates on the same element of the $\mathbb{Q}(n,k)$-vector space of normal-form elements, with basis the $f_{n+i,k+j}$ for (i,j) in U.

For example, when $f_{n,k}$ is the binomial coefficient $\binom{n}{k}$, the rules (8) instantiate as the two relations

$$\binom{n+1}{k} = \frac{n+1}{n+1-k}\binom{n}{k}, \quad \binom{n}{k+1} = \frac{n-k}{k+1}\binom{n}{k}. \tag{9}$$

Rewriting with them in this case, (7) would reduce to an identity between rational functions, after factoring out $\binom{n}{k}$, which computer algebra should easily prove or disprove. However, the nullity of rational functions relies on relations like $(n-k)/(n-k) = 1$, which, *if the rational functions have to be interpreted as functions* and not just algebraic fractions, are only valid for $n \neq k$. When it comes to general ∂-finite sequences f, the phenomenon is the same, with the only difference that normal forms do not allow to factor out a single term like $\binom{n}{k}$ above, but gather rational functions in front of the $f_{n+i,k+j}$ for (i,j) in U.

Therefore, the best that this approach can do is prove identity (7) outside of an algebraic locus. Yet, the existing algorithms in computer algebra prove nothing more than the algebraic interpretation of (7), and do not return any constraint for validity.

Moreover, by focusing on a theory of Gröbner bases for recurrence operators over a rational-function field, working with ∂-finite sequences relies on the idea that a recurrence can without loss be symbolically multiplied by a rational function. But, are the two recurrences

$$n\,(u_{n+1} - u_n) = 0 \qquad \text{and} \qquad u_{n+1} - u_n = 0$$

really equivalent? In fact no, as the solutions to the second are only the constant sequences, while the solutions to the first also contain sequences that are nonzero at 0 and zero for $n \geq 1$. If one is to encode such a nonconstant sequence by the second recurrence, then the recurrence has to be decorated by the proviso $n \neq 0$, which goes out of the algebraic theory of Gröbner bases. Of course, the phenomenon persists for sequences in more indices.

In cases like deriving asymptotic properties of univariate sequences, the algebraic treatment by computer algebra, with no determination of a first integer at which the "proved" identity holds, may be sufficient. But another phenomenon occurs in summation by creative telescoping, whether it be by Zeilberger's fast algorithm [41] or Chyzak's algorithm [8]. Indeed, the common approach evaluates multivariate rational functions after normalizing them, potentially disregarding functions like $(n-k)/(n-1)$, whose value at $k = n = 1$ depends on the ordering of taking limits.

A possible work-around known in the computer-algebra literature [36] is to introduce a new variable ϵ and work with $k + \epsilon$ instead of k, so as to avoid integer values. But this generates equations of larger orders and total sizes that will not permit the computations to scale, even before considering manipulating them in a proof assistant.

In the present work, we exhibit no cases where the ∂-finite approach leads to false proofs of wrong identities. Rather, we suggest that it can provide incomplete proofs of valid identities. In fact, *we doubt* sufficiently to fear false proofs.

4. SOUND CREATIVE TELESCOPING

The main result of this section is Lemma 1 below, which is the crux of the proof of recurrences obtained by closure under (definite) summation, like for a, s, and b in (6): given a bivariate sequence $(u_{n,k})_{(n,k)\in\mathbb{Z}^2}$ to be summed into a definite sum $(U_n)_{n\in\mathbb{Z}}$, it formalizes the general process of deriving a recurrence relation for U from a creative-telescoping recurrence relation on the summand u, namely (10) below. We state the lemma over any field K, for future work, but for this work on $\zeta(3)$, only $K = \mathbb{Q}$ will be used.

One of the hypotheses often used in the literature to derive the simple conclusion (4) is that the summation has *standard boundary conditions*, that is, that the sum for U_n is over all values of k that make the summand $u_{n,k}$ nonzero, all other values being (defined and) zero. (More generality is possible, but goes beyond the scope of this presentation.) For example, $\sum_{k=0}^{n}\binom{n}{k}$ has standard boundary conditions, while $\sum_{k=0}^{n}\binom{2n}{k}$ and $\sum_{k=0}^{n-1}\binom{n}{k}/(n-k)$ do not. In Lemma 1 below, equation (11) is more involved, to accommodate potentially nonstandard boundary conditions.

We stress that, in the next lemma, all evaluations are supposed to be well-defined, and the sequences and operators to be total. Other situations are discussed after the lemma.

Lemma 1 *Let $r \in \mathbb{N}$, and, for $0 \leq i \leq r$, let $p_i \in K^{\mathbb{Z}}$ be a function. Introduce P, the linear operator defined by $(Py)_n = \sum_{i=0}^{r} p_i(n)\, y_{n+i}$ for any univariate sequence y and $n \in \mathbb{Z}$. Let Q be an operator on bivariate sequences. Consider $u \in K^{\mathbb{Z}^2}$ and integers α and β from $\mathbb{Z}$, and let U be the sequence with general term $U_n = \sum_{k=\alpha}^{n+\beta} u_{n,k}$. Then, for any set $\Delta \subset \mathbb{Z}^2$ for which*

$$(n,k) \notin \Delta \Rightarrow (Pu_{\_,k})_n = (Qu)_{n,k+1} - (Qu)_{n,k}, \tag{10}$$

the following identity holds for any n such that $\alpha \leq n + \beta$:

$$\begin{aligned}(PU)_n = {} & (Qu)_{n,n+\beta+1} - (Qu)_{n,\alpha} \\ & + \sum_{\substack{\alpha\leq k\leq n+\beta \\ (n,k)\in\Delta}} (Pu_{\_,k})_n - (Qu)_{n,k+1} + (Qu)_{n,k} \\ & + \sum_{i=1}^{r}\sum_{j=1}^{i} p_i(n) u_{n+i,n+\beta+j}.\end{aligned} \tag{11}$$

The proof of identity (11) is a straightforward reordering of the terms of the left-hand side $(PU)_n = \sum_{i=0}^{r} p(n)\, U_{n+i}$ after unfolding the definition of U and applying relation (10) everywhere allowed in the interval $\alpha \leq k \leq n + \beta$. In other

words, a starting point to derive (11) is to sum (10) over this interval, then compensate for the cases $(n,k) \in \Delta$. The first part of the right-hand side is the usual difference of border terms. The last part of the right-hand side is the collection of terms that arise from the fact that the upper bound of the sum defining U_n depends linearly on n and that we do not assume any nullity of the summand outside the summation domain. The middle part of the right-hand side, which we will call the singular part, witnesses the possible partial domain of validity of relation (10).

Equations (10) and (11) above are formal, generalized forms for (3) and (4) in our simplistic sketch in Section 2.2. In the litterature, too, only the first part of the right-hand side of (11) is given, both because of simplifying assumptions (that are however not satisfied on many examples) or because of incomplete proofs. See, e.g., Theorem 5.1 as well as Section 6.3 in [42], the sentence including equation (2) in [43], the one-line proof of the Fundamental corollary in [38], Theorem 1 and Corollary A in [39], equation (7.1.5) in [25], and, for honesty sake, in the work of an author of the present article, the simplifying hypothesis (3.1) leading to equation (3.4) in [11], and the similar treatment in [8, Section 3]. Rare exceptions that treat nonstandard boundary conditions with some level of generality, both in the context of multiple sums, are the works [36, Sec. 3.4] and [32, Chapter 3] (applied in [7]). The latter does not provide a formula, but a procedure to write down analogues of (11) in the case of complicated boundary conditions.

A variant of (11) allows a more direct comparison with how it is usually stated: in

$$\begin{aligned}(PU)_n &= (Qu)_{n,n+b+r+1} - (Qu)_{n,a} \\ &\quad + \sum_{\substack{a\le k\le n+b+r \\ (n,k)\in\Delta}} (Pu_{\_,k})_n - (Qu)_{n,k+1} + (Qu)_{n,k} \\ &\quad - \sum_{i=0}^{r}\sum_{j=i+1}^{r} p_i(n) u_{n+i,n+b+j}\end{aligned}$$

the third part vanishes in the context of a summand that is zero outside the bounds of summation (*natural boundaries*) and the singular part is also overlooked.

Note that we do not assume linearity of Q. This allows for its use, for example, in the context of creative telescoping in difference extensions [31].

Observe that the collection of singular terms crucially depends on the definition of the set Δ restricting the creative-telescoping identity: the larger Δ is, the more difficult it will be in practice to simplify the complete expression. In the extreme but unrealistic case where Δ is $\mathbb{Z}^2$, Lemma (1) becomes totally uninformative.

Nevertheless, this set Δ can be put to good use to deal with two kinds of singularities that appear in practice: "sequences" in applications need not be total functions (e.g., $\binom{n}{k}/(n-k)$ mentioned above); expressions for "operators" produced in practice by creative-telescoping algorithms often feature rational functions that prevent their direct interpretation as total functions from sequences to sequences. In both cases, salvation comes from prolonging the rational functions in the expressions as piecewise-defined functions with some arbitrary values at their singular loci. This way, a partial sequence $\tilde{u}$ defining a sum $U_n = \sum_{k=\alpha}^{n+\beta} \tilde{u}_{n,k}$ in a well-defined manner, and partial operators $\tilde{P}$ and $\tilde{Q}$ that would not be amenable to Lemma 1 are replaced with prolongations u, P, and Q, for which the lemma applies. Barring unlikely coincidence, the loci of such prolongations contribute to Δ (repeated and shifted in the proper way).

Prolonging is implicit in the Coq library we used [2], as it declares the inverse of the rational number 0 to be 0, a simple way of ensuring that all functions are total. As a counterpart, all formal lemmas involving rational inverse or division take this redefinition into account in their premises, e.g., by enforcing that all denominators are nonzero.

Because of the interpretation of the recurrences (8) that define a ∂-finite sequence as a Gröbner basis, it has been customary in the literature to favour homogeneous recurrences, and thus, variants of (11) with null right-hand side. Such a variant always exists as a consequence of (11), as computer algebra provides algorithms to look for an operator P' cancelling the right-hand side, thus leading by composition to $(P'PU)_n = 0$. We have enforced that our Maple script directly return pairs (P,Q) with this nullity property.

A pair (P,Q) is called a *creative-telescoping pair*. A relation of the shape (10) is called a *creative-telescoping identity*.

5. FORMAL PROOFS OF RECURRENCES

Our formal proof that the sequences a and b in (1) satisfy the same second-order recurrence (2) follows program (5): we prove a collection of lemmas that formalize the results obtained by algorithmic calculations and we apply these lemmas to the sequences defined in (6). This proves that a is a solution of (2) and that b is a solution of some recurrence of order four. We conclude that the recurrence (2) holds for b as well by using evaluations of this sequence.

In all what follows we use the names introduced in (6) for specific sequences. Variables with hats ($\hat{z}$, $\hat{c}$, ...) denote arbitrary sequences. Each capital letter in program (5) refers both to a system of recurrences and to the characteristic function of its set of solutions. For instance, $C(c)$ should be read "the system C holds for the sequence c".

5.1 Recurrences with provisos

In our formal proof, a recurrence is defined as a conditional equation. The proviso makes explicit the values of the indices at which an instance of the equation is well defined and holds. For instance the leaf system C of program (5) is the conjunction of two bivariate, first-order, conditional recurrences:

$$(n,k) \notin \Delta_1 \Rightarrow \hat{c}_{n+1,k} = \left(\frac{n+1}{n+1-k}\right)^2 \hat{c}_{n,k}, \tag{12}$$

$$(n,k) \notin \Delta_2 \Rightarrow \hat{c}_{n,k+1} = \left(\frac{(n-k)(n+1+k)}{k+1}\right)^4 \hat{c}_{n,k}, \tag{13}$$

with $\Delta_1 := \{(n,k) : n = -1 \wedge k = n+1\}$ and $\Delta_2 := \{(n,k) : n = 0 \wedge k+1 = 0\}$.

A skeptical use of computer–algebra calculations requires executing program (5) two times to complete the formal proof. The first run consists in executing a Maple script which computes one system of—unconditional—recurrence equations per inner node in (5). The script pretty-prints its output in Coq syntax and generates empty placeholders for the provisos. For example, in the case of system C, the Maple script generates two formulae matching exactly statements (12) and (13), including references to Δ_1 and Δ_2.

The values of these two provisos are then written down by hand in the Coq script as described in Section 5.2.

In order to complete the formal proof, we run the program a second time inside the proof assistant. This second run consists in proving one lemma per inner node in (5). For instance, at the inner node V we prove that:

$$\forall \hat{c} \in \mathbb{Q}^{\mathbb{Z}^2},\ \forall \hat{u} \in \mathbb{Q}^{\mathbb{Z}^2},\ C(\hat{c}) \wedge U(\hat{u}) \Rightarrow V(\hat{c} \times \hat{u}) \qquad (14)$$

where the sequence $\hat{c} \times \hat{u} \in \mathbb{Q}^{\mathbb{Z}^2}$ is the pointwise product of the sequences $\hat{c}$ and $\hat{u}$. Similarly, at node B we prove that:

$$\forall \hat{v} \in \mathbb{Q}^{\mathbb{Z}^2},\ V(\hat{v}) \Rightarrow B\left(\sum_{k=0}^{n} \hat{v}_{n,k}\right). \qquad (15)$$

Note that those lemmas are not specific to the sequences defined in (6). For instance formula (14) states that the recurrences V hold not only for v but for any choice of a sequence $\hat{v}$ satisfying the premise systems C and U. However, we indeed prove that each sequence in (6) is a solution of the eponymous *conditional* system in the program and in particular that A, resp. B, holds for a, resp. b.

5.2 Sources of provisos

Provisos should at least exclude the possible poles of the fractions involved in the equations. Yet in most cases they are even more restrictive.

The concrete sequences in (6) may for instance not satisfy the recurrence systems for all values of their indices. The provisos annotating the leaf systems Z, C, and D are hence designed so as both to exclude the poles of the coefficients and to provide sufficient conditions under which $Z(z)$, $C(c)$, and $D(d)$ hold respectively.

But finding appropriate provisos at inner nodes of the program is more intricate. We start with a complete and definitive definition of the premise systems, like C and U in the case of (14) and V in the case of (15). In particular, the values of their provisos have been devised at earlier stages of the formal proof. We also have a candidate, proviso-free system of equations for the result of the closure, which has been calculated by the Maple program. Validating the closure operation requires crafting an appropriate set of restrictions for this candidate system. These restrictions should at least: exclude the poles of the coefficients, make the system be satisfied by the eponymous sequence in (6), but also allow for an a posteriori proof of the closure lemma.

Even more challengingly, the different lemmas of the program cannot be considered independently. For instance, oversizing the restrictions in system V cripples the proof of statement (15). As a result, appropriate values for the provisos can hardly be anticipated solely from the equations calculated by Maple and definitions in (6). In fact it is difficult to guess these values without a first trial run of the proof, in order to discover the correct obligations.

5.3 A complete proof of the program

We prove each lemma of the formal program by using the method sketched in Section 3. However a notable difference is that the analogues of relations (8), which we use to normalize a candidate identity, now feature provisos. For each node of the program, the Maple script generates a system of recurrences that is a Gröbner basis. But their annotation with conditions jeopardizes the normalization strategy of the proviso-free case: provisos may indeed exclude some of the reduction paths and even compromise the confluence of the reduction.

For each addition and product of sequences, we prove a statement analogous to (14). We proceed exactly as suggested in Section 3: we normalize a candidate identity (7) with respect to known—but now conditional—relations and conclude by comparing to zero the rational-function coefficients of the remaining terms. We verify that each step in the simplification is a legal instance of one of the rules. Values for which the identity cannot be proved under the conditional rules are excluded by the hand-crafted proviso annotating the conclusion.

For each definite summation, we prove a statement analogous to (15). We use Lemma 1 to validate the recurrences computed by creative-telescoping algorithms, from the creative-telescoping pairs produced by the Maple script. For each creative-telescoping pair, we use the set of rules known on the summand to verify a creative-telescoping identity (10). This verification follows the same protocol as the one we described for addition and product, including the discovery of a correct proviso.

The rest of the proof consists in: applying Lemma 1 to this creative-telescoping identity, then normalizing the right-hand side of the instance of equation (11) obtained this way. As mentioned at the end of Section 4, thanks to the calculations performed by our Maple script, we expect that this expression normalizes to zero. This expression has a more general form than the ones that we have dealt with so far to validate recurrences for addition and product, and creative-telescoping identities. First, we should verify that the prolongations we introduced to interpret ill-defined rational functions compensate, so that in the end the expression to be normalized does not depend on them. The resulting expression features several distinct collections of terms that will normalize to zero by independent simplification chains. The upper border term and the overhead terms belong to the same collection since they are all shifts from a same origin term, with index $(n, n+\beta)$. The lower border term and the singular terms contribute to other distinct collections of terms, obtained by shifts from different origins. For instance in our running example (15), we observe that after the cancelling of prolongations, the expression to be normalized features two distinct collections: a finite set of shifts from $\hat{v}_{n,n}$ and a finite set of shifts from $\hat{v}_{n,0}$. We observe that the latter does not only contain the border term $\hat{v}_{n,0}$ but also several singular terms. The simplification of the extra collections of terms may in principle require additional assumptions about some specializations of the summand. Yet by design, our Maple script generates no such extra assumption. It happens that we have been able to verify completely all the recurrences obtained by creative telescoping with this protocol. However we have no guarantee that on other examples we would be able to complete such a posteriori proofs without strengthening the assumptions on the summand.

5.4 Formal proofs and automation

The provisos we use have been recorded by hand in the Coq script. We have not tried to maintain tight conditions but rather concise and readable ones. For instance, we sometimes anticipate on a sign constraint caused by the range of a definite sum in the program, which may subsume some large conjunction of equalities to negative values.

We however use a formal-proof-producing decision procedure [5, Chapter 21] to justify that each reduction modulo a conditional recurrence that we perform is legal. Each of these proof obligations is a first-order formula in the theory of integer arithmetics (with order). Although this theory is undecidable, the formulae we deal with all fall in a fragment that is amenable to automated decision. In practice, we enforce these proof obligations to be stated as satisfiability problems and the polynomials in the atoms to be products of linear factors. This latter feature, crucial to efficiency, is easily realized by having the Maple script enforce an appropriate factorization discipline in the coefficients of the recurrences pretty-printed in Coq files.

In the end, our formal proof does not depend on the Maple script that has discovered the recurrences. Verifying the conjectures produced by the script nonetheless requires performing some calculations inside the logic underlying the proof assistant, namely normalizations of rational functions. The rational functions involved in this proof are rather small with respect to the standards of computer algebra systems but already challenging for proof assistants. The approach to formal certification adopted in this work benefits from the status of computation in the logic underlying the Coq proof assistant [12, 13]. This meta-theoretical feature goes so far as to allow optimizations of the system implementation that make computations quite efficient [17]. We heavily use this feature of Coq when calling the formal-proof-producing decision procedure [18][34, Chapter 24] that normalizes rational functions automatically.

6. CONCLUSIONS AND PERSPECTIVES

Formalizing beyond the recurrences.

In the present article, we have focused on the part of our formal development that addresses the proof of Apéry's recurrence (2). But we also formalized the other parts of Apéry's proof as sketched in Section 2.1, following a mix of Salvy's text [29] and van der Poorten's report [35]. We rely on existing broad libraries of formalized mathematics [2] that accommodate the variety of arithmetic and analytic objects involved. Still, at some places, we have used more elementary variants of the proofs in [35, 29]. We avoid this way the need for sophisticated general theories that are not available in state-of-the-art libraries of formalized mathematics. More details will be found in our upcoming [10].

At the time of writing, our formal proof of irrationality is strictly speaking not complete. Indeed, it uses the assertion that $\ell_n = \mathcal{O}(3^n)$ without providing any machine-checked formal proof of this (known) fact. Note that this result is totally independent from the rest of the irrationality proof. Hence, again strictly speaking, we provide a complete formal proof of the statement:

$$\ell_n = O(3^n) \Rightarrow \zeta(3) \notin \mathbb{Q}$$

We plan to machine-check Apéry's proof completely, by formalizing the proof proposed by Hanson [19].

Variant algorithmic proof paths.

Variants of the efficient algorithms we have used in the present proof allow a similar algorithmic approach to the proof of Apéry's recurrence, possibly working with different intermediate recurrences and therefore different sets of singularities. But we think that these alternative approaches suffer from a similar incompleteness to ours'.

For instance one could use Wilf–Zeilberger pairs [38] to compute closures by definite summation, as this produces recurrences with polynomial coefficients instead of rational functions. However this appealing feature is a lure since in this case as well the validation of the creative-telescoping pair relies on the normalization of *rational* functions.

Another interesting alternative proposed by Schneider allows to verify directly that recurrence (2) holds for both a and b, when the approach we follow uses an intermediate recurrence of degree four for b. This would remove the part of the proof devoted to reducing the order of the latter, using initial conditions for b. But a counterpart is to deal with nested sums and a priori more involved provisos.

Still, it would be interesting to apply our formal study to these alternative proof paths.

Completeness of our approach.

It would be even more satisfying to better understand on which class of problems the methods of the previous paragraph are actually sound. This would make it possible to revisit computer-algebra algorithms to re-develop them in a "safe mode".

This insight would open the way for an automation of the formal proofs that validate recurrences, to the point that both a formal statement and its formal proof are entirely generated from a Maple script.

Acknowledgments

We wish to thank Enrico Tassi for helping us in the formalization effort and Alin Bostan for clarifying some arithmetic proofs and for pointing us to [19, 15].

7. REFERENCES

[1] Algolib. http://algo.inria.fr/libraries/, 2013. Version 17.0. For Maple 17.

[2] Mathematical Components Libraries. http://www.msr-inria.fr/projects/mathematical-components, 2013. Version 1.4. For Coq 8.4pl3.

[3] R. Apéry. Irrationalité de $\zeta(2)$ et $\zeta(3)$. *Astérisque*, 61, 1979. Société Mathématique de France.

[4] R. Apéry. Interpolation de fractions continues et irrationalité de certaines constantes. In *Mathematics*, CTHS: Bull. Sec. Sci., III, pages 37–53. Bib. Nat., Paris, 1981.

[5] F. Besson. Fast reflexive arithmetic tactics: the linear case and beyond. In *Types for proofs and programs*, volume 4502 of *Lecture Notes in Comput. Sci.*, pages 48–62. Springer, Berlin, 2007.

[6] F. Beukers. A note on the irrationality of $\zeta(2)$ and $\zeta(3)$. *Bull. London Math. Soc.*, 11(3):268–272, 1979.

[7] J. Blümlein, S. Klein, C. Schneider, and F. Stan. A symbolic summation approach to Feynman integral calculus. *J. Symbolic Comput.*, 47(10):1267–1289, 2012.

[8] F. Chyzak. An extension of Zeilberger's fast algorithm to general holonomic functions. *Discrete Math.*, 217(1-3):115–134, 2000.

[9] F. Chyzak, M. Kauers, and B. Salvy. A non-holonomic systems approach to special function identities. In

ISSAC'09: Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation, pages 111–118. ACM, New York, 2009.

[10] F. Chyzak, A. Mahboubi, and E. Tassi. A computer-algebra based formal proof of the irrationality of $\zeta(3)$. In the process of being submitted to ITP'2014, 2014.

[11] F. Chyzak and B. Salvy. Non-commutative elimination in Ore algebras proves multivariate identities. *J. Symbolic Comput.*, 26(2):187–227, 1998.

[12] T. Coquand and G. P. Huet. The calculus of constructions. *Inf. Comput.*, 76(2/3):95–120, 1988.

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

[14] J. A. De Loera, J. Lee, P. N. Malkin, and S. Margulies. Hilbert's Nullstellensatz and an algorithm for proving combinatorial infeasibility. In *ISSAC 2008*, pages 197–206. ACM, New York, 2008.

[15] B.-y. Feng. An simple elementary proof for the inequality $d_n < 3^n$. *Acta Math. Appl. Sin. Engl. Ser.*, 21(3):455–458, 2005.

[16] S. Gerhold and M. Kauers. A procedure for proving special function inequalities involving a discrete parameter. In *ISSAC'05*, pages 156–162. ACM, New York, 2005.

[17] B. Grégoire and X. Leroy. A compiled implementation of strong reduction. In *International Conference on Functional Programming 2002*, pages 235–246. ACM Press, 2002.

[18] B. Grégoire and A. Mahboubi. Proving equalities in a commutative ring done right in Coq. In *Theorem proving in higher order logics*, volume 3603 of *Lecture Notes in Comput. Sci.*, pages 98–113. Springer, Berlin, 2005.

[19] D. Hanson. On the product of the primes. *Canad. Math. Bull.*, 15:33–37, 1972.

[20] J. Harrison and L. Théry. A skeptic's approach to combining HOL and Maple. *J. Automat. Reason.*, 21(3):279–294, 1998.

[21] M. Kauers. Computer proofs for polynomial identities in arbitrary many variables. In *ISSAC 2004*, pages 199–204. ACM, New York, 2004.

[22] M. Kauers, C. Koutschan, and D. Zeilberger. Proof of Ira Gessel's lattice path conjecture. *Proc. Natl. Acad. Sci. USA*, 106(28):11502–11505, 2009.

[23] C. Koutschan, M. Kauers, and D. Zeilberger. Proof of George Andrews's and David Robbins's q-TSPP conjecture. *Proc. Natl. Acad. Sci. USA*, 108(6):2196–2199, 2011.

[24] C. Koutschan and T. A. Thotsaporn. Advanced computer algebra for determinants. *Ann. Comb.*, 17(3):509–523, 2013.

[25] M. Petkovšek, H. S. Wilf, and D. Zeilberger. $A = B$. A K Peters Ltd., Wellesley, MA, 1996.

[26] V. Pillwein. Termination conditions for positivity proving procedures. In *ISSAC 2013*, pages 315–321. ACM, New York, 2013.

[27] T. Rivoal. La fonction zêta de Riemann prend une infinité de valeurs irrationnelles aux entiers impairs. *C. R. Acad. Sci. Paris Sér. I Math.*, 331(4):267–270, 2000.

[28] T. Rivoal. *Propriétés diophantiennes de la fonction zêta de Riemann aux entiers impairs*. PhD thesis, Université de Caen, 2001.

[29] B. Salvy. An Algolib-aided version of Apéry's proof of the irrationality of $\zeta(3)$. http://algo.inria.fr/libraries/autocomb/Apery2-html/apery.html, 2003.

[30] B. Salvy and P. Zimmermann. Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Trans. Math. Software*, 20(2):163–177, 1994.

[31] C. Schneider. A refined difference field theory for symbolic summation. *J. Symbolic Comput.*, 43(9):611–644, 2008.

[32] F. Stan. *Algorithms for special functions: computer algebra and analytical aspects*. PhD thesis, RISC, 2010.

[33] R. P. Stanley. Differentiably finite power series. *European J. Combin.*, 1(2):175–188, 1980.

[34] The Coq development team. The Coq proof assistant: reference manual. http://coq.inria.fr/refman/, 2013. Version v8.4pl3.

[35] A. van der Poorten. A proof that Euler missed: Apéry's proof of the irrationality of $\zeta(3)$. *Math. Intelligencer*, 1(4):195–203, 1979. An informal report.

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

[37] H. S. Wilf and D. Zeilberger. Towards computerized proofs of identities. *Bull. Amer. Math. Soc. (N.S.)*, 23(1):77–83, 1990.

[38] H. S. Wilf and D. Zeilberger. An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/integral identities. *Invent. Math.*, 108(3):575–633, 1992.

[39] H. S. Wilf and D. Zeilberger. Rational function certification of multisum/integral/"q" identities. *Bull. Amer. Math. Soc. (N.S.)*, 27(1):148–153, 1992.

[40] D. Zeilberger. Identities. In *q-series and partitions (Minneapolis, MN, 1988)*, volume 18 of *IMA Vol. Math. Appl.*, pages 35–44. Springer, New York, 1989.

[41] D. Zeilberger. A fast algorithm for proving terminating hypergeometric identities. *Discrete Math.*, 80(2):207–211, 1990.

[42] D. Zeilberger. A holonomic systems approach to special functions identities. *J. Comput. Appl. Math.*, 32(3):321–368, 1990.

[43] D. Zeilberger. The method of creative telescoping. *J. Symbolic Comput.*, 11(3):195–204, 1991.

[44] D. Zeilberger. Identities in search of identity. *Theoret. Comput. Sci.*, 117(1-2):23–38, 1993. Conference on Formal Power Series and Algebraic Combinatorics (Bordeaux, 1991).

[45] V. V. Zudilin. One of the numbers $\zeta(5)$, $\zeta(7)$, $\zeta(9)$, $\zeta(11)$ is irrational. *Uspekhi Mat. Nauk*, 56(4(340)):149–150, 2001.