Computer Proofs for Polynomial Identities in Arbitrary many Variables

Manuel Kauers*
Research Institute for Symbolic Computation
Johannes Kepler Universität Linz
A-4040 Linz
mkauers@risc.uni-linz.ac.at

ABSTRACT

We present an algorithm for proving certain families of polynomial identities in which the number of variables appears as a parameter. Typically, indefinite sums and products (possibly nested) are used for defining such polynomial identities, but in principle the definition of the problem may be given in terms of arbitrary polynomial recurrences (possibly nonlinear). An implementation of our algorithm in Mathematica was able to verify identities appearing in textbooks, which, to our knowledge, were previously not accessible by any symbolic method. A collection of examples is included in the paper.

Categories and Subject Descriptors

I.1.2 [**Computing Methodologies**]: Symbolic and Algebraic Manipulation—*Algorithms*

General Terms

Algorithms

Keywords

Polynomial identities, computer proofs

1. INTRODUCTION

Plenty of automated tools have been successfully applied to handle various sorts of problems with a fixed number of variables. Zeilberger's pioneering work [14, 13], for instance, has meanwhile been successfully extended to an amazing generality [3, 2]. However, all this is still concerned with a fixed number of variables only. There are almost no techniques available for the case where the number of unknowns is itself unknown. In [15], Zeilberger pointed out that the next major step to be done is the development of algorithms for proving identities in which the number of variables itself is unspecified. The goal of this paper is to contribute to the this subject, albeit at a very modest level of polynomial identities.

It has been suggested in [9, 15] to employ the theory of symmetric functions (see, e.g., [11, 7]) for proving polynomial identities of this kind. For example, the identity

$$\Big(\sum_{i=1}^{n} x_i\Big)^2 = \sum_{i=1}^{n} x_i^2 + 2 \sum_{1\leq i<j\leq n} x_i x_j \tag{1}$$

can be expressed in terms of the classical symmetric polynomials

$$p_1(n) = \sum_{i=1}^{n} x_i, \quad p_2(n) = \sum_{i=1}^{n} x_i^2, \quad e_2(n) = \sum_{i=1}^{n} \Big(x_i \sum_{j=1}^{i-1} x_j\Big)$$

as $p_1(n)^2 = p_2(n) + 2e_2(n)$. Stembridge's package SF [12] can be used to automatically prove this identity by bringing $-p_1(n)^2 + p_2(n) + 2e_2(n)$ to a normal form with respect to, e.g., the basis $e_1, e_2, \ldots$ Indeed, the command `toe(-p1^2+p2+2*e2)` for doing this returns 0, thus proving identity (1) for arbitrary n.

A different approach for proving identities in arbitrary many variables, applicable to a much wider class of objects, is presented in this paper. The algorithm we will describe is an adaption of the algorithm for proving classical identities in one discrete indeterminate which we presented in [6]. This algorithm proceeds by computing a natural number k such that the identity is true for all $n \in \mathbb{N}$ if and only if it holds for the first k values. Already in the original paper we mentioned that the algorithm can be applied to problems involving a sequence $(x_n)_{n=1}^{\infty}$ of indeterminates. However, as opposed to the new version presented here, the variation given in [6] does not terminate in general.

Both, the original algorithm and the modified version presented here, are quite elementary applications of basic notions of difference algebra [4]. We will briefly introduce the necessary concepts of this field in Section 2. Then, in Section 3, we review the results of [6]. The main contribution of the current paper is a generalization of these results. This generalization, which can be applied to prove a wide range of polynomial identities in an arbitrary number of variables,

*Partially supported by the Austrian Science Foundation (FWF) grant F1305 and by the German Academic Exchange Service (DAAD) grant D/03/40515.

is presented in Sections 4 and 5. A collection of simple examples, which have been successfully tackled by an implementation of our algorithm in a Mathematica package, is provided in Sections 4.4 and 5.3. The examples mainly serve illustrative purposes and are indeed also not hard to prove by hand. However, it is noteworthy that no other known symbolic method applies to any of them.

2. BASIC NOTIONS FROM DIFFERENCE ALGEBRA

We introduce some basic concepts from difference algebra. Our notation deviates only slightly from the original notation used in [4].

A commutative ring $\mathfrak{B}$ together with a distinguished automorphism $s\colon \mathfrak{B} \to \mathfrak{B}$ is called a *difference ring.* The automorphism s is called the *shift operator* on $\mathfrak{B}$. Difference rings, which are actually fields, are called *difference fields.* If $\mathfrak{B}$ is a difference ring and $a \in \mathfrak{B}$, then we also write $sa, s^2a, \dots$ for $s(a), s(s(a)), \dots$.

Let $\mathfrak{F}$ be a difference field, $m \in \mathbb{N}$. By $\mathfrak{F}\{t_1, t_2, \dots, t_m\}$ we denote the m-fold *polynomial difference ring* over $\mathfrak{F}$ consisting of all polynomials in $t_i, st_i, s^2t_i, \dots$ $(i = 1, \dots, m)$ with coefficients in $\mathfrak{F}$. The set $\{\, s^it_j : i \in \mathbb{N}_0, j = 1, \dots, m \,\}$ of formal shifts is algebraically independent. The elements of $\mathfrak{F}\{t_1, \dots, t_m\}$ are called *difference polynomials.* The *order* of a difference polynomial p is the maximum number r such that s^rt_i appears in p for some i. The quotient field $Q(\mathfrak{F}\{t_1, \dots, t_m\})$ is denoted by $\mathfrak{F}\langle t_1, \dots, t_m\rangle$. The notations $\mathfrak{F}[t_1, \dots, t_m]$ and $\mathfrak{F}(t_1, \dots, t_m)$ refer to the usual polynomial ring in $t_1, \dots, t_m$ and its quotient field, respectively.

We view $\mathfrak{F}\{t_1, \dots, t_m\}$ as a polynomial ring in infinitely many variables. By $\mathfrak{F}\{t_1, \dots, t_m\}_k$ $(k \in \mathbb{N})$ we denote the kth *restriction* of $\mathfrak{F}\{t_1, \dots, t_m\}$, defined as the set of all difference polynomials $p \in \mathfrak{F}\{t_1, \dots, t_m\}$ whose order is at most k. $\mathfrak{F}\{t_1, \dots, t_m\}_k$ is viewed as a polynomial ring over $\mathfrak{F}$ in $m(k+1)$ variables. Its quotient field is consequently denoted by $\mathfrak{F}\langle t_1, \dots, t_m\rangle_k$. This latter object is closely related to what is called *difference kernel* in [4].

Now, consider the difference ring $\mathfrak{B} := \mathfrak{F}\{t_1, \dots, t_m\}$ for some difference field $\mathfrak{F}$. For a set $S \subseteq \mathfrak{B}$, $\langle S\rangle$ denotes the polynomial ideal generated by S in $\mathfrak{B}$ (written (S) in [4]). By $\langle\!\langle S\rangle\!\rangle$ we denote the *difference ideal* of S, defined as the smallest polynomial ideal of $\mathfrak{B}$ containing S and being closed under s (written $[S]$ in [4]). We define $\langle S\rangle_k := \langle S\rangle \cap \mathfrak{B}_k$ and $\langle\!\langle S\rangle\!\rangle_k := \langle\!\langle S\rangle\!\rangle \cap \mathfrak{B}_k$. Note that $\langle S\rangle_k$ and $\langle\!\langle S\rangle\!\rangle_k$ are themselves ideals in $\mathfrak{B}_k$. However, $\langle\!\langle S\rangle\!\rangle_k$ is of course not a difference ideal because the restriction $\mathfrak{B}_k$ is not a difference ring. Also note that $\langle\!\langle S\rangle\!\rangle_k$ is finitely generated, and that by

$$\langle\!\langle S\rangle\!\rangle_k = \langle S \cup sS \cup \dots \cup s^kS\rangle \cap \mathfrak{B}_k$$

we can easily obtain a finite ideal basis for $\langle\!\langle S\rangle\!\rangle_k$ from a finite basis of $\langle S\rangle_k$.

3. IDENTITIES IN ONE FREE VARIABLE

In this section, we briefly summarize the results of [6]. By $\mathfrak{F}$ we denote an algebraically closed field of characteristic zero. We will first define the class $\mathcal{C}$ of *nested polynomially recurrent* sequences. Then we will show how zero equivalence can be decided for this class.

3.1 Admissible Sequences

We consider the class $\mathcal{C} = \mathcal{C}(\mathfrak{F})$ of sequences in $\mathfrak{F}$. $\mathcal{C}$ is inductively defined as follows. If $f_1, \dots, f_m \in \mathcal{C}$ $(s \geq 0)$ and if $(f(n))_{n=1}^{\infty}$ is a sequence in $\mathfrak{F}$ which obeys a recurrence of the form

$$f(n+r) = p \qquad \text{or} \qquad f(n+r) = 1/p \tag{2}$$

where p is a polynomial expression in $f_1(n)$, $f_1(n+1)$, ..., $f_1(n+r)$, $f_2(n)$, ..., $f_2(n+r)$, ..., $f_m(n)$, ..., $f_m(n+r)$ and $f(n), \dots, f(n+r-1)$ with coefficients in $\mathfrak{F}$. The number r is called the *order* of the recurrence. Note that the base of the induction is given for $s = 0$.

We call $\mathcal{C}$ the class of *nested polynomially recurrent* sequences, reflecting the facts that the defining recurrence of one sequence may involve other sequences (the $f_1, \dots, f_m$ above), and that the polynomial p may involve nonlinear terms in f like $f(n)f(n+1)$.

As pointed out in [6], the class $\mathcal{C}$ contains plenty of frequently appearing sequences, e.g., holonomic sequences, and it enjoys some useful closure properties. For instance, $\mathcal{C}$ is closed under field operations and under taking indefinite sums, products, and continued fractions. In addition to common sequences, $\mathcal{C}$ also contains such peculiar sequences as $(\mathrm{Fib}(2^n))_{n=1}^{\infty}$ or $(2^{\mathrm{Fib}(n)})_{n=1}^{\infty}$ where $\mathrm{Fib}(\cdot)$ denotes the Fibonacci sequence (or, more generally, any sequence obeying a C-finite recurrence; see, e.g., [14], Section 2.1.2 for a definition of C-finite).

3.2 The Zero Equivalence Test in $\mathcal{C}$

The elements of $\mathcal{C}$ can be modeled by a finitely generated difference ideal of a multivariate polynomial difference ring. The recurrences of a sequence $f \in \mathcal{C}$ and the sequences occurring in the definition of f (transitively) form a suitable basis: If the $f_1, \dots, f_m$ are represented by the variables $t_1, \dots, t_m$ and f itself is to be represented by t_{m+1} then the defining recurrence (2) translates into the basis elements

$$b := s^rt_{m+1} - p(\dots) \quad \text{or} \quad b := s^rt_{m+1}p(\dots) - 1,$$

respectively, where "..." stands for the variables

$$\begin{array}{ccccc} t_1, & st_1, & \cdots & s^{r-1}t_1, & s^rt_1, \\ t_2, & st_2, & \cdots & s^{r-1}t_2, & s^rt_2, \\ \vdots & \vdots & & \vdots & \vdots \\ t_m, & st_m, & \cdots & s^{r-1}t_m, & s^rt_m, \\ t_{m+1}, & st_{m+1}, & \cdots & s^{r-1}t_{m+1}. & \end{array}$$

The polynomial b is called the *defining relation* of t_m. The $t_1, \dots, t_m$ have defining relations of the same type, possibly depending on indeterminates $t_i, st_i, \dots$ with smaller i for which in turn defining relations are available. The important fact is that every t_i has a defining relation depending only on t_j with $j < i$ (possibly shifted).

Given a set $D \subseteq \mathfrak{F}\{t_1, \dots, t_m\}_r$ of defining relations for $t_1, \dots, t_m$ where t_m represents the sequence $f = (f(n))_{n=1}^{\infty} \in \mathcal{C}$ as described above, the algorithm proceeds as follows.

function isZero(D, f)
  $I_r := \langle\!\langle D \rangle\!\rangle_r + \langle t_m, st_m, \ldots, s^{r-1}t_m\rangle_r$
  $k := r$
  while $s^k t_m \notin \mathrm{Rad} I_k$ **do**
    $k := k + 1$
    $I_k := \langle\!\langle D \rangle\!\rangle_k + \langle t_m, st_m, \ldots, s^{k-1}t_m\rangle_k$
  done
  if $f(1) = f(2) = \cdots = f(k) = 0$ **then**
    return true
  else
    return false

In the listing above, we assume that t_m corresponds to f. $\mathrm{Rad} I_k$ refers to the radical ideal of I_k. Ideal computations are taking place in appropriate restrictions of the difference ring $\mathfrak{F}\{t_1, \ldots, t_m\}$.

Detailed proofs of correctness and termination of this algorithm are given in [6]. In fact, both correctness and termination will also follow as special case from the statements made in the following section.

4. THE CASE OF ARBITRARY MANY VARIABLES

The algorithm outlined in Section 3 is now applied to identities in unknowns $x_1, x_2, \ldots, x_n$ for undetermined n. The key idea is to regard $x_1, x_2, \ldots$ as a sequence without defining relation, represented like the other sequences by a formal variable t_i and its formal shifts.

4.1 Admissible Sequences

The class $\mathcal{C}' = \mathcal{C}'(\mathfrak{F})$ of formal sequences in $\mathfrak{F}$ is defined inductively by:

1. Formal sequences of the form $u_1, u_2, u_3, \ldots$ belong to the class $\mathcal{C}'$. By a formal sequence, we mean a sequence of distinct symbols bearing no relation to each other.
2. If $f_1, \ldots, f_s \in \mathcal{C}'$ $(s \geq 0)$ and $f = (f(n))_{n=1}^{\infty}$ obeys a recurrence of the form (2), then $f \in \mathcal{C}'$.

Obviously, $\mathcal{C} \subseteq \mathcal{C}'$. Also note that for every $f \in \mathcal{C}'$ and every $n \in \mathbb{N}$, $f(n)$ evaluates to a rational function in $\mathfrak{F}(u_1, \ldots, u_n)$.

At this point we remark that $\mathcal{C}'$ contains in particular the classical symmetric polynomials p_1, p_2, e_2 used in the introduction for proving (1). Appropriate defining relations are readily found as

$$\begin{aligned} p_1(n+1) &= p_1(n) + x_{n+1} \\ p_2(n+1) &= p_2(n) + x_{n+1}^2 \\ e_2(n+1) &= e_2(n) + x_{n+1}p_1(n). \end{aligned}$$

Additional, "more complicated" objects in $\mathcal{C}'$ with interesting applications are discussed in Section 4.4.

4.2 The Zero Equivalence Test in $\mathcal{C}'$

We can treat elements of $\mathcal{C}'$ in the same way as we treated the elements of $\mathcal{C}$ in Section 3, omitting any defining relation for the variables corresponding to formal sequences $u_1, u_2, \ldots$ There are instances where the algorithm still terminates as it stands, and if it does so, we can perfectly rely on the answer. But the existence of defining relations for *every* t_i is crucial for the termination, and so we cannot expect termination of the original algorithm in general. As an example, let $u_1, u_2, \ldots$ be arbitrary and define $(p_n)_{n=1}^{\infty}$, $(q_n)_{n=1}^{\infty}$ by the recurrences

$$\begin{aligned} p_{n+2} &= u_{n+2}p_{n+1} + p_n, \quad p_{-2} = 0, p_{-1} = 1, \\ q_{n+2} &= u_{n+2}q_{n+1} + q_n, \quad q_{-2} = 1, q_{-1} = 0. \end{aligned}$$

Then it turns out that the algorithm as stated in Section 3 is able to prove

$$p_{n+1}q_n - p_n q_{n+1} = (-1)^n$$

in a finite number of steps, but it can be shown that it does not terminate on

$$p_{n+1}q_{n-1} - p_{n-1}q_{n+1} = (-1)^{n+1}u_{n+1}.$$

(Both identities are special cases of (35) in I.§6 of [8].)

In order to avoid this nontermination due to missing defining relations of variables representing arbitrary sequences $(u_n)_{n=1}^{\infty}$, we put these "critical" variables into the ground field. If $t_1, \ldots, t_{i-1}$ are the variables corresponding to formal sequences $u_1, u_2, \ldots$ then the algorithm of Section 3 carries over literally, if we stipulate that the I_k are ideals in $\mathfrak{F}\langle t_1, \ldots, t_{i-1}\rangle\{t_i, \ldots, t_m\}_k$ rather than in $\mathfrak{F}\{t_1, \ldots, t_m\}_k$.

4.3 Correctness and Termination

We will next prove correctness and termination of this modification. The proofs are very similar to those already given in [6] for the case where $\mathfrak{F}$ contains only constants. We leave out some technical details which can be found there.

THEOREM 1. *The algorithm is correct.*

PROOF. If the algorithm returns **false,** then it has found some $i \in \{1, \ldots, k\}$ with $f(i) \neq 0$ which provides a counterexample.

On the other hand, if the algorithm returns **true,** we may prove the correctness by induction on n. In this case, the induction hypothesis $f(1) = \cdots = f(k) = 0$ is fulfilled. As for the induction step, we observe that $\langle\!\langle D \rangle\!\rangle_k$ has by construction (see [6] for a detailed formal argument) the following property $(*)$: For every $j, n \in \mathbb{N}$, every point in the variety of

$$\langle\!\langle D \rangle\!\rangle_j + \langle t_m - f(n), st_m - f(n+1), \ldots, s^{j-1}t_m - f(n+j-1)\rangle_j$$

(i.e., every solution of the corresponding system of polynomial equations) satisfies $s^j t_m = f(n+j)$. By induction hypothesis we have $f(n) = \cdots = f(n+k-1) = 0$, and the termination condition of the main loop asserts

$$s^k t_m \in \mathrm{Rad}(\langle\!\langle D \rangle\!\rangle_k + \langle t_m, \ldots, s^{k-1}t_m\rangle_k).$$

Together with $(*)$ this implies that $f(n+k) = 0$. $\square$

The next lemma provides a statement which is required later for the termination proof. The claim of the lemma can be shown by elementary algebraic arguments. A detailed proof is also given in [6].

Lemma 2. *Let K be an arbitrary field of characteristic 0 and $K[X] = K[x_1, \dots, x_n]$ be the polynomial ring in n variables over K. Let P be a prime ideal in $K[X]$ and $q \in K[X]$.*

Then the ideals $P' := \langle P \cup \{y-q\}\rangle$ and $P'' := \langle P \cup \{qy-1\}\rangle$ in $K[X, y]$ are also prime and the quotient fields of their coordinate rings are isomorphic:

$$Q(K[X,y]/P') \cong Q(K[X]/P) \cong Q(K[X,y]/P'').$$

Theorem 3. *The algorithm terminates.*

Proof. We observe that

$$I_{k+1} = \langle I_k \cup \{s^k t_m,\ s^{k-r+1} p_i, \dots, s^{k-r+1} p_m\}\rangle_{k+1}$$

where p_j is the defining relation of t_j ($j = i, \dots, m$).

If $s^k t_m \in \mathrm{Rad} I_k$, then we are done. If not, then there must be a primary component P of I_k with $s^k t_m \notin P$. By primality, $\langle P \cup s^k t_m\rangle_k$ has strictly lower dimension as $\langle P\rangle_k$. It follows that $I_k + \langle s^k t_m\rangle_k$ has at least one component replaced by components of strictly lower dimensions, compared to I_k.

Every defining relation is either of the form $s^{r_j} t_j q_j - 1$ or $s^{r_j} t_j - q_j$ for some polynomial q_j depending only on $t_{j'}$ with $j' \le j$ (and shifts). Hence m-fold application of Lemma 2 asserts that $J := I_k + \langle s^k t_m\rangle_k$ and

$$I_{k+1} = \langle J \cup \{s^{k-r+1} p_1, \dots, s^{k-r+1} p_m\}\rangle_{k+1}$$

have the same number of primary components and corresponding components have the same dimension. Therefore, in each iteration, at least one component of I_k is replaced by components of strictly lower dimension. By Dixon's lemma, this process must ultimately come to an end. $\square$

4.4 Examples and Applications

We illustrate how the algorithm proves the identity

$$\sum_{k=1}^{n} \frac{\prod_{i=1}^{k-1}(x_i + \alpha)}{\prod_{i=1}^{k} x_i} = \frac{1}{\alpha}\Big(\prod_{k=1}^{n} \frac{x_k + \alpha}{x_k} - 1\Big) \qquad (3)$$

taken from Exercise 5.93 in [5]. We take $\mathfrak{F} = \mathbb{Q}(\alpha)$ as field of constants. There is one sequence of variables, $(x_n)_{n=1}^{\infty}$, which will be represented by t_0, i.e., t_0 corresponds to x_n, st_0 to x_{n+1}, etc. Further difference indeterminates are used for representing the sum and the products in terms of defining relations. We compute in the ring $\mathbb{Q}(\alpha)\langle t_0\rangle\{t_1, \dots, t_6\}$ and use the following set D of defining relations:

$$\begin{aligned}
D := \{&st_1 - t_1(t_0 + \alpha), && (\,t_1 \sim \Pi(x_i + \alpha)\,)\\
&st_2 - t_2 st_0, && (\,t_2 \sim \Pi(x_i)\,)\\
&t_3 t_2 - 1, && (\,t_3 \sim 1/\Pi(x_i)\,)\\
&st_4 - t_4 - st_1 st_3, && (\,t_4 \sim \mathrm{lhs}\,)\\
&st_5 - t_5(st_0 + \alpha)/st_0, && (\,t_5 \sim \Pi(x_k + \alpha)/x_k\,)\\
&t_6 - t_4 + (t_5 - 1)/\alpha\,\} && (\,t_5 \sim \mathrm{lhs} - \mathrm{rhs}\,)
\end{aligned}$$

A quick Gröbner basis computation confirms that $st_6 \notin \mathrm{Rad}(\langle\!\langle D\rangle\!\rangle_1 + \langle t_6\rangle_1)$ but $sst_6 \in \mathrm{Rad}(\langle\!\langle D\rangle\!\rangle_2 + \langle t_6, st_6\rangle_2)$, therefore we obtain $k = 2$. Now, it only remains to check (3) for $n = 1$ and $n = 2$, which is straightforward.

To give an impression of the range of examples to which our method can be applied, we list some more example identities which we were able to prove by a straightforward implementation of our algorithm in Mathematica. The implementation also supports certain transformations from expressions to sets of defining relations, which makes it possible to enter the problems quite similar to the form in which they are stated in this paper. The implementation is available from the author's homepage.

All problems listed here, including the detailed example above, are completed in less than a second on a 2.4 GHz machine with 1Gb of memory, using the built-in Gröbner basis facilities of Mathematica.

As the second example, consider the Christoffel-Darboux formula for orthogonal polynomials [1, Theorem 5.2.4]. If $(\lambda_n)_{n=1}^{\infty}$ and $(c_n)_{n=1}^{\infty}$ denote arbitrary sequences, then the family $(P_n(x))_n$ of orthogonal polynomials, defined by

$$\begin{aligned}
&P_n(x) = (x - c_n)P_{n-1}(x) - \lambda_n P_{n-2}(x),\\
&P_{-1}(x) = 0, \quad P_0(x) = 1,
\end{aligned}$$

satisfies the formula

$$\sum_{k=0}^{n} \frac{P_k(x)P_k(u)}{\prod_{i=1}^{k+1}\lambda_i} = \frac{P_{n+1}(x)P_n(u) - P_n(x)P_{n+1}(u)}{(x-u)\prod_{i=1}^{n+1}\lambda_i}$$

for all $n \in \mathbb{N}$. Note that in this example the λ_n and c_n play the role of the unknowns that appear in an undetermined number while x and u are just regarded as constants. Note further that we have not just proven the identity for some special family of orthogonal polynomials, but rather for all families of orthogonal polynomials in one stroke.

A related example which can be proven just as fast as the Christoffel-Darboux identity is taken from the theory of continued fractions. If $K(y_0, \dots, y_n; x_1, \dots, x_n)$ denotes the continuant polynomial ([5, Section 6.7] or [8]) then

$$C_n := x_0 + \cfrac{y_1}{x_1 + \cfrac{y_2}{\ddots + \cfrac{y_n}{x_n}}} = \frac{K(y_1, \dots, y_n; x_0, \dots, x_n)}{K(y_2, \dots, y_n; x_1, \dots, x_n)}$$

may serve as a representation for the continued fraction by which the computer proves instantaneously the identity

$$C_n = x_0 + \sum_{k=1}^{n}(-1)^{k+1}\frac{\prod_{i=1}^{k} y_i}{B_{k-1}B_k}$$

in full generality for all $n \in \mathbb{N}$ and all $x_0, x_1, \dots, y_1, y_2, \dots$ (Exercise 5.12 in [1]). Here, B_λ abbreviates the denominator's continuant polynomial $K(y_2, \dots, y_\lambda; x_1, \dots, x_\lambda)$.

At the end of this section, let us us stress that proving identities like the following two artificial ones now has become completely routine for the computer. These identities are generalizations of identities which have been found by the Sigma package [10] with particular sequences in place of the x_n. The generalization has been found by hand, based on Sigma's answers for special instances.

The symbol $\mathrm{H}(n) = \sum_{k=1}^{n} 1/k$ in (4) denotes the nth harmonic number.

$$\sum_{k=1}^{n} \mathrm{H}(k) \sum_{i=1}^{k} x_i = (-n + (1+n)\mathrm{H}(n)) \sum_{i=1}^{n} x_i$$
$$- \sum_{k=1}^{n} (\mathrm{H}(k)-1) k x_k\,, \tag{4}$$

$$\sum_{r=1}^{n} \frac{\sum_{l=1}^{r} \prod_{k=1}^{l} \frac{x_k}{1+x_k}}{1+x_r}$$
$$= \sum_{k=1}^{n} \frac{\prod_{i=1}^{k} \frac{x_i}{1+x_i}}{1+x_k} + \sum_{i=1}^{n} \frac{1}{1+x_i} \sum_{i=1}^{n} \prod_{k=1}^{i} \frac{x_k}{1+x_k}$$
$$- \sum_{k=1}^{n} \Big(\prod_{i=1}^{k} \frac{x_i}{1+x_i} \Big) \Big(\sum_{i=1}^{k} \frac{1}{1+x_i} \Big) \tag{5}$$

The left and right hand side of (4) and (5) do not define symmetric functions, as can be seen already for $n = 2$. Hence, these identities are out of the scope of the symmetric functions approach outlined in the introduction.

5. GENERALIZATION: APPLYING MULTIVARIATE IDENTITIES

It has been shown in the previous section how polynomial identities in $x_1, \ldots, x_n$, $y_1, \ldots, y_n$, etc. can be proven automatically for arbitrary n. Under certain circumstances, there is a simple extension of the methodology of the previous section which applies also to identities in $x_1, \ldots, x_n$, $y_1, \ldots, y_m$, etc. for arbitrary and independent n, m, etc. This will be outlined next.

5.1 Admissible Sequences

We define the class $\mathcal{C}^* = \mathcal{C}^*(\mathfrak{F})$ of multivariate formal sequences in $\mathfrak{F}$ inductively by: $\mathcal{C}'(\mathfrak{F}) \subseteq \mathcal{C}^*(\mathfrak{F})$ and the object $(f(n_1, \ldots, n_p))_{n_1,\ldots,n_p=1}^{\infty}$ belongs to $\mathcal{C}^*(\mathfrak{F})$ if and only if $(f(n_1, n_2, \ldots, n_p))_{n_1=1}^{\infty} \in \mathcal{C}'(\mathfrak{F}(n_2, \ldots, n_p))$ and for every value $\alpha \in \mathbb{N}$, the object $(f(\alpha, n_2, \ldots, n_p))_{n_2,\ldots,n_p=1}^{\infty}$ belongs to $\mathcal{C}^*(\mathfrak{F})$.

Note that the order of variables is significant. For example, $f(m, n, k) = k^{2^n + m}$ belongs to $\mathcal{C}^*$, but $f(k, m, n) = k^{2^n + m}$ does not.

5.2 Zero Equivalence Test in $\mathcal{C}^*$

Given a sequence $f = (f(n_1, \ldots, n_p))_{n_1,\ldots,n_p=1}^{\infty} \in \mathcal{C}^*$, we proceed as follows to answer the question $f \overset{?}{\equiv} 0$. First consider f as an element of $\mathcal{C}'(\mathfrak{F}(n_2, \ldots, n_p))$, and compose a set of defining relations for f and its components. This is possible by definition of $\mathcal{C}'$. The algorithm of Section 4 is then applied to compute a number k of initial values that needs to be checked in order to get a decision. For checking these initial values, proceed recursively with $(f(\alpha, n_2, \ldots, n_p))_{n_2,\ldots,n_p=1}^{\infty}$, for $\alpha = 1, \ldots, k$. Note that the recursion is again possible by definition of $\mathcal{C}^*$.

5.3 Two Simple Examples

The procedure just described can be used to do Exercise 5.27 in [5] which asks the reader to prove the hypergeometric identity

$$F\left(\begin{matrix} a_1, a_1 + \frac{1}{2}, \ldots, a_m, a_m + \frac{1}{2} \\ b_1, b_1 + \frac{1}{2}, \ldots, b_n, b_n + \frac{1}{2}, \frac{1}{2} \end{matrix} \,\Big|\, (2^{m-n-1} z)^2 \right)$$
$$= \frac{1}{2} \left[F\left(\begin{matrix} 2a_1, \ldots, 2a_m \\ 2b_1, \ldots, 2b_n \end{matrix} \,\Big|\, z \right) + F\left(\begin{matrix} 2a_1, \ldots, 2a_m \\ 2b_1, \ldots, 2b_n \end{matrix} \,\Big|\, -z \right) \right] \tag{6}$$

for arbitrary $m, n \geq 0$. Recall the definition

$$F\left(\begin{matrix} a_1, \ldots, a_m \\ b_1, \ldots, b_n \end{matrix} \,\Big|\, z \right) := \sum_{k \geq 0} \frac{\prod_{i=1}^{m} a_i^{\overline{k}}}{\prod_{i=1}^{n} b_i^{\overline{k}}} \frac{z^k}{k!},$$

where $a^{\overline{k}} = a(a+1)\cdots(a+k-1)$ denotes the raising factorial.

The odd coefficients of z^k in (6) obviously cancel out, and by consideration of the even coefficients it suffices to prove

$$\frac{2^{2k(m-n-1)}(2k)!}{(1/2)^{\overline{k}}} \frac{\prod_{i=1}^{m} a_i^{\overline{k}} (a_i + \frac{1}{2})^{\overline{k}}}{\prod_{i=1}^{n} b_i^{\overline{k}} (b_i + \frac{1}{2})^{\overline{k}}} = \frac{\prod_{i=1}^{m} (2a_i)^{\overline{2k}}}{\prod_{i=1}^{n} (2b_i)^{\overline{2k}}} \tag{7}$$

for all $n, m, k \geq 0$. Induction on m requires one to check two initial values, $m = 0$ and $m = 1$, and for both of these special values of m, induction on n also requires to check two initial values, $n = 0, n = 1$. Thus it remains to show the identity for the four special cases $(m, n) = (0,0)$, $(0,1)$, $(1,0)$, $(1,1)$ for all $k \geq 0$. These identities are in a form to which the algorithm of Section 4 can be applied. Altogether, about five seconds of CPU time were needed to complete the proof of (7) in full generality.

As another example, consider the identity

$$K(x_1, \ldots, x_{m+n})$$
$$= K(x_1, \ldots, x_m) K(x_{m+1}, \ldots, x_{m+n}) \tag{8}$$
$$+ K(x_1, \ldots, x_{m-1}) K(x_{m+2}, \ldots, x_{m+n})$$

(Eq. (6.133) in [5]) for the special continuant polynomials $K(x_1, \ldots, x_n) := K(1, \ldots, 1; x_1, \ldots, x_n)$. First, we keep m fixed and put $u_n := x_{m+n}$. (This substitution was done by hand.) Then the algorithm of Section 4 is applied to

$$K(u_{1-m}, \ldots, u_n) = \alpha K(u_1, \ldots, u_n) + \beta K(u_2, \ldots, u_n)$$

with constants α, β. It turns out that three initial conditions have to be checked: $n = 1, 2, 3$. Plugging these values of n into (8) gives three identities in m that can again be tackled by the algorithm of Section 4. The overall computation time needed to complete this example is far less than a second.

6. CONCLUSION

We have presented an algorithm for proving, or refuting, identities in one discrete indeterminate n. Our algorithm applies commutative algebra techniques, in particular the radical membership test, to appropriate restrictions of a difference ideal representing the identity under consideration, and thereby determines a number of values which suffices to be checked in order to conclude that the identity holds everywhere. It has been shown that this algorithm can be successfully applied to the problem of proving, or refuting, polynomial identities in an undetermined number of variables. The input class of the presented algorithm includes in particular the class of symmetric polynomials for which automated proofs were already available. Indeed, it covers identities of a far more general type, including also sequences obeying nonlinear recurrences of arbitrary order, not necessarily yielding symmetric functions.

Now, does this fulfill Zeilberger's wish of being able to prove identities in arbitrary many variables? Actually, not quite. In his outlook [15], he aims at being able to deal with objects like $(\sum_{i=1}^{n} x_i)!$ or $\prod_{i=1}^{n} \binom{2n_i}{n_i}$ which are beyond the scope of our approach. However, he does suspect that in analogy to the classical WZ-machinery [9], algorithms for proving polynomial identities might be needed as a subroutine. Our algorithm might be a contribution at this level.

Directions for more modest applications have been sketched in Sections 4.4 and 5.3. Certain kinds of problems having been accessible to symbolic methods only for special instances so far can now be handled in full generality by the computer. For the given examples, this is even possible very efficiently.

Acknowledgements
I would like to thank Peter Paule for suggesting the topic of this paper. I am also thankful to the anonymous referees for numerous suggestions that helped improving the clarity of the paper.

7. REFERENCES

[1] G. E. Andrews, R. Askey, and R. Roy. *Special Functions*, volume 71 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1999.

[2] F. Chyzak. An extension of Zeilberger's fast algorithm to general holonomic functions. *Discrete Mathematics*, 217:115–134, 2000.

[3] F. Chyzak and B. Salvy. Non-commutative elimination in Ore algebras proves multivariate identities. *Journal of Symbolic Computation*, 26:187–227, 1998.

[4] R. M. Cohn. *Difference Algebra*. Interscience Publishers, John Wiley & Sons, 1965.

[5] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, second edition, 1994.

[6] M. Kauers. An algorithm for deciding zero equivalence of nested polynomially recurrent sequences. Technical Report 2003-48, SFB F013, Johannes Kepler Universität, 2003. (submitted).

[7] I. G. MacDonald. *Symmetric Functions and Orthogonal Polynomials*. American Mathematical Society, 1998.

[8] O. Perron. *Die Lehre von den Kettenbrüchen*. Chelsea Publishing Company, second edition, 1929.

[9] M. Petkovšek, H. Wilf, and D. Zeilberger. $A = B$. AK Peters, Ltd., 1997.

[10] C. Scheider. *Symbolic Summation in Difference Fields*. PhD thesis, RISC-Linz, Johannes Kepler Universität Linz, 2001.

[11] R. P. Stanley. *Enumerative Combinatorics, Volume 2*. Cambridge Studies in Advanced Mathematics 62. Cambridge University Press, 1999.

[12] J. R. Stembridge. A Maple package for symmetric functions. *Journal of Symbolic Computation*, 20:755–768, 1995.

[13] D. Zeilberger. A fast algorithm for proving terminating hypergeometric identities. *Discrete Mathematics*, 80:207–211, 1990.

[14] D. Zeilberger. A holonomic systems approach to special functions. *Journal of Computational and Applied Mathematics*, 32:321–368, 1990.

[15] D. Zeilberger. WZ Theory, Chapter II. In *MSRI Workshop on Symbolic Computation in Geometry and Analysis*, 1998. Transcript available at http://www.math.rutgers.edu/~zeilberg/.