Holonomic Systems and Automatic Proofs of Identities

Frédéric Chyzak
INRIA and École Polytechnique

October 3, 1994

[summary by Bruno Salvy]

Abstract

D. Zeilberger has shown how many combinatorial identities involving special functions can be proved using the theory of holonomic sequences and functions. This work presents a general algorithmic approach to the multivariate case, together with an implementation.

Introduction

Speaking informally, D. Zeilberger has defined holonomic functions in [10] as those functions of one or several variables satisfying sufficiently many *linear* equations (differential equations or recurrence relations) with polynomial coefficients so that they are completely determined by a finite number of initial conditions and a finite number of polynomial coefficients. The study of these functions is motivated by their pervasiveness in combinatorics and special functions theory. The class of holonomic functions enjoys closure properties that make it possible to construct equations satisfied by a particular function from equations satisfied by simpler functions. These operators can be exploited by many algorithms. In particular, series expansions of holonomic function can be computed efficiently and asymptotic estimates related to them can be derived from the operators. A very important special class of holonomic functions is formed by algebraic functions, for which finding a differential operator is the best known algorithm to compute series expansions of large order.

For the single variable case, the *gfun* package [6] provides functions that construct recurrence or differential operators satisfied by holonomic sequences or functions and thus prove formulæ. For instance, Cassini's identity on the Fibonacci numbers

$$F_{n+2}F_n - F_{n+1}^2 = (-1)^n,$$

is proved by computing a linear recurrence satisfied by the left-hand side, starting from the linear recurrence satisfied by the Fibonacci numbers. Here is the kind of proof for which *gfun* provides tools:

$$\begin{aligned} h_n &= F_{n+2}F_n - F_{n+1}^2 = F_n^2 + F_nF_{n+1} - F_{n+1}^2 \\ h_{n+1} &= F_{n+1}^2 + F_{n+1}F_{n+2} - F_{n+2}^2 = F_{n+1}^2 - F_{n+1}F_n - F_n^2 = -h_n. \end{aligned}$$

It is then sufficient to check that $(-1)^n$ also satisfies this recurrence and that a finite number of initial conditions match.

Ore operator	$\sigma(x)$	$\delta(x)$	Action
differentiation	x	1	$f(x) \mapsto f'(x)$
shift	$x+1$	0	$f(x) \mapsto f(x+1)$
difference	$x+1$	1	$f(x) \mapsto f(x+1) - f(x)$
q-dilation	qx	0	$f(x) \mapsto f(qx)$
q-differentiation	qx	1	$f(x) \mapsto [f(qx) - f(x)]/[(q-1)x]$
Mahlerian operator	x^p	0	$f(x) \mapsto f(x^p)$

TABLE 1. Examples of Ore operators

In one variable, the algorithms for differential equations and recurrence equations are always very similar and can be profitably expressed using the vocabulary of *Ore operators* [5]. These are defined over a field $K(x)$ by a commutation rule

$$\partial x = \sigma(x)\partial + \delta(x), \tag{1}$$

where σ is a ring endomorphism of $K(x)$ and δ is a vector-space endomorphism of $K(x)$. Table 1 gives a list of important examples.

In several variables, a holonomic function is defined by several operators and most of the closure properties still hold. In addition, holonomy is often preserved by specialization; by definite and indefinite summation (for recurrences) or integration (for differential equations). However, making the corresponding construction of operators explicit is more difficult. Wilf and Zeilberger [9] have given efficient algorithms for some of these operations in the *hypergeometric* and the *q-hypergeometric* case (linear recurrences or q-recurrences of order 1 on a sequence or on the coefficients of a series). N. Takayama [7, 8] has used Gröbner bases of differential, difference and q-difference operators to make an explicit construction of operators in the general (non-hypergeometric) case. This is (at least partly) implemented in his programs *Kan* and *Macaulay for D-modules.*

The aim of this work is to attack multivariate holonomy via Ore operators and non-commutative elimination by Gröbner bases or a skew version of the Euclidean algorithm [1, 2]. This is implemented in F. Chyzak's *Mgfun* package[1] written in Maple.

1. Elimination and ideals

Ore algebras in the univariate case are algebras $K\langle x, \partial\rangle$ where K is a ring and x and ∂ are related by (1). The multivariate case is obtained as the tensor product $K\langle x_1, \partial_1\rangle \otimes \cdots \otimes K\langle x_n, \partial_n\rangle$.

The example of Legendre polynomials illustrates a simple use of elimination. Legendre polynomials satisfy the following three dependent relations:

$$(1-x^2)P_n''(x) - 2xP_n'(x) + n(n+1)P_n(x) = 0, \tag{2}$$

$$(n+2)P_{n+2}(x) - (2n+3)xP_{n+1}(x) + (n+1)P_n(x) = 0, \tag{3}$$

$$(1-x^2)P_{n+1}'(x) + (n+1)xP_{n+1}(x) - (n+1)P_n(x) = 0. \tag{4}$$

Any of these relations can be deduced from the other ones. Here is how *Mgfun* can be used to prove (3) from (2) and (4). The computation consists in defining a suitable Ore algebra, a proper term ordering on the variables, and then computing a Gröbner basis with respect to this order.

[1]Available by anonymous ftp on `ftp.inria.fr:INRIA/Projects/algo/programs/Mgfun` or at the URL `http://www-rocq.inria.fr/Combinatorics-Library/www/programs/Mgfun`.

```
A:=orealg([x,diff,Dx],[n,shift,Sn]): T:=termorder(A,plex=[Dx,Sn],max):
DE:=(1-x^2)*Dx^2-2*x*Dx+n*(n+1): RDE:=(1-x^2)*Dx*Sn+(n+1)*x*Sn-(n+1):
map(collect,gbasis([DE,RDE],T,ratpoly(rational,[n,x])),Sn,factor);
```

$$[(-n-1)S_n + xn - D_x + x^2 D_x + x, \quad (-n-2)S_n^2 + x(2n+3)S_n - n - 1]$$

The operator (3) appears at the end of the basis. The same result can be obtained by a skew Euclidean algorithm applied to (2) and (4). This is done in *Mgfun* as follows:

```
RE:=skewelim(DE,RDE,Dx,A,ratpoly(rational,[n,x])):
```

Since we are interested in functions or sequences annihilated by operators, it is natural to consider the left ideal generated by these operators. In one variable, the ring $K(x)\langle\partial\rangle$ is Euclidean (therefore principal) [5]. Thus one can work with solutions of univariate Ore operators as one works with algebraic numbers, using Euclid's algorithm to compute normal forms in a finite dimensional vector space generated by $1, \partial, \partial^2, \ldots, \mathcal{P}$ where $\mathcal{P}$ is an operator analogous to the minimal polynomial. These normal forms in turn are used to compute operators annihilating sums or products of holonomic functions by performing computations in the proper finite dimensional vector space and determining a linear relation by Gaussian elimination. In several variables, skew polynomial rings are Noetherian [3] so that a normal form is provided by Gröbner bases. The same kind of algorithms as in the univariate case apply. Elimination between operators consists in finding an element of the ideal they generate which does not contain the undesirable variable. A recursive extended gcd algorithm can be used to eliminate a ∂ variable between two operators. The general case of elimination is obtained by Gröbner bases with appropriate orders.

2. Creative telescoping

Holonomic ideals form an important class of ideals of operators. In these ideals, it is possible to eliminate *any* of the variables. This elimination is applied by *creative telescoping* [11] to the computation of definite integrals or sums. The idea is that if f is annihilated by a holonomic ideal of $K\langle x,\partial\rangle$ and if the $\partial^k f$'s ($k \geq 0$) vanish at the border $\partial\Omega$ of a suitable domain Ω, then an operator annihilating $\partial_\Omega^{-1} f$ (the definite sum or integral) is obtained by first eliminating x. This yields an operator which can be rewritten $\partial A(\partial) + B$ such that

$$[\partial \cdot A(\partial)](f) + B(f) = 0.$$

Then applying ∂_Ω^{-1} (i.e., summing or integrating over the domain) gives $A(\partial)(f)|_{\partial\Omega} + \partial_\Omega^{-1} B(f) = 0$, where the hypotheses ensures that the first part (the sum or integral at the boundary) is zero. Since B does not contain x it commutes with ∂^{-1} and is the desired operator.

As an example, we compute a system of differential equations satisfied by the generating function of the Legendre polynomials

$$F(x,z) = \sum_{n\geq 0} P_n(x) z^n$$

starting from (2) and (3). The steps to be performed are: *(i)* creation of the Ore algebra $\mathcal{A} = \mathbb{Q}\langle n, S_n\rangle \otimes \mathbb{Q}\langle x, \partial_x\rangle \otimes \mathbb{Q}\langle z, \partial_z\rangle$; *(ii)* determination of operators annihilating $P_n(x)z^n$ in $\mathcal{A}$; *(iii)* elimination of n; *(iv)* left division by $S_n - 1$. Here is the corresponding *Mgfun* session:

```
Legendre:=[RE,DE,Dz]: z_to_the_n:=[Dx,z*Dz-n,Sn-z]:
A:=orealg([x,diff,Dx],[n,shift,Sn],[z,diff,Dz]):
T1:=termorder(A,tdeg=[Dx,Sn,Dz],max):
Legendre_times_zn:=hprod(Legendre,z_to_the_n,2,T1);
```

Legendre_times_zn := $[D_x^2 - x^2 D_x^2 - 2xD_x + n^2 + n, nS_n^2 + 2Sn^2 + z^2 n + z^2 - 2zxnS_n - 3zxS_n, zD_z - n]$

```
T2:=termorder(A,lexdeg=[[n],[Dx,Sn,Dz]],max):
gb:=gbasis(Legendre_times_zn,T2,ratpoly(rational,[x,n,z]));
```

$$gb := [D_x^2 - x^2D_x^2 - 2xD_x + z^2D_z^2 + 2zD_z, z - xS_n + S_n^2D_z - 2zxS_nD_z + z^2D_z, zD_z - n]$$

```
map(collect,subs(Sn=1,[gb[1],gb[2]]),[Dx,Dz]);
```

$$[(1 - x^2)D_x^2 - 2xD_x + 2zD_z + z^2D_z^2, (1 - 2zx + z^2)D_z + z - x]$$

This is a system of differential equations satisfied by the generating function of the Legendre polynomials. The whole computation is performed in 2.1 s. on a Dec Alpha. The system can be further simplified by another elimination to yield an operator of second order in D_x only. From the above system a symbolic solver of differential equations can be used to find the well-known formula

$$\sum_{n\geq 0} P_n(x)z^n = \frac{1}{\sqrt{1 - 2xz + z^2}}.$$

Conclusion

This approach is susceptible to numerous applications, extensions and improvements. Applications to q-computations look promising; Comtet's algorithm [4] to compute the differential equation satisfied by an algebraic function can be generalized to some extent; a program handling operators and initial conditions simultaneously could benefit from the initial conditions to avoid letting the orders of the operators grow too much and thus could turn into an efficient formulæ prover; computation of Gröbner bases could be speeded up using a non-commutative analogue of trace lifting or simple generalizations of the FGLM algorithm, etc. Hopefully, all of this will appear in F. Chyzak's thesis.

Bibliography

[1] Chyzak (Frédéric). – *Holonomic Systems and Automatic Proofs of Identities.* – Research Report n° 2371, Institut National de Recherche en Informatique et en Automatique, October 1994.

[2] Chyzak (Frédéric) and Salvy (Bruno). – Non-commutative elimination in Ore algebras proves multivariate holonomic identities. – In preparation, 1995.

[3] Cohn (P. M.). – *Free Rings and Their Relations.* – Academic Press, 1971, *London Mathematical Society Monographs.*

[4] Comtet (L.). – Calcul pratique des coefficients de Taylor d'une fonction algébrique. *L'Enseignement Mathématique*, vol. 10, 1964, pp. 267–270.

[5] Ore (Oystein). – Theory of non-commutative polynomials. *Annals of Mathematics*, vol. 34, 1933, pp. 480–508.

[6] Salvy (Bruno) and Zimmermann (Paul). – Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Transactions on Mathematical Software*, vol. 20, n° 2, 1994, pp. 163–177.

[7] Takayama (Nobuki). – Gröbner basis, integration and transcendental functions. In *Symbolic and algebraic computation.* ACM, pp. 152–156. – 1990. Proceedings ISSAC'90, Kyoto.

[8] Takayama (Nobuki). – An approach to the zero recognition problem by Buchberger algorithm. *Journal of Symbolic Computation*, vol. 14, 1992, pp. 265–282.

[9] Wilf (Herbert S.) and Zeilberger (Doron). – An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/integral identities. *Inventiones Mathematicae*, vol. 108, 1992, pp. 575–633.

[10] Zeilberger (Doron). – A holonomic systems approach to special functions identities. *Journal of Computational and Applied Mathematics*, vol. 32, n° 3, 1990, pp. 321–368.

[11] Zeilberger (Doron). – The method of creative telescoping. *Journal of Symbolic Computation*, vol. 11, 1991, pp. 195–204.