Holonomic Functions in Mathematica

Christoph Koutschan
Johann Radon Institute for Computational and Applied Mathematics
Austrian Academy of Sciences
Altenberger Straße 69, A-4040 Linz, Austria
`christoph.koutschan@ricam.oeaw.ac.at`

Abstract

We present the Mathematica package HolonomicFunctions which provides a powerful framework for the automatic manipulation of multivariate holonomic functions, in the spirit of Zeilberger's holonomic systems approach. Its top-level functionalities are: converting a mathematical expression into a holonomic description, executing holonomic closure properties, and creative telescoping for general holonomic functions. To achieve these goals, many other, lower-level, functionalities had to be implemented which were not available in the Mathematica system: finding rational solutions of linear systems of (q-) difference / differential equations, noncommutative arithmetic in Ore algebras and computing Gröbner bases in such domains.

In his seminal paper *A holonomic systems approach to special functions identities* [11], Zeilberger writes: "I hope that more professional programmers and algorithmic designers will soon expand the rudimentary ideas in this paper and develop a symbolic software package to prove general special function identities." Meanwhile, his dream has certainly become true: numerous papers refining and extending his "rudimentary ideas" have appeared, new and faster algorithms have been developed (some of them by Zeilberger himself), and there are several software packages available which implement (at least parts of) his holonomic systems approach.

Our package HolonomicFunctions is definitely one of the most general ones, since it allows to "prove general special function identities", as anticipated by Zeilberger. As special cases, our package contains his fast algorithm for proving hypergeometric identities [10] and its q-analogue, as well as the Almkvist-Zeilberger algorithm [1] for evaluating integrals of hyperexponential functions. Moreover, it can likewise deal with sums (resp. integrals) of functions which don't satisfy first-order recurrences (resp. differential equations), but higher-order ones. The only other software package we know of, providing this degree of generality, is Chyzak's Mgfun package for Maple. Our main motivations to reimplement the algorithms already contained in Mgfun were 1) to make them available to Mathematica users, 2) to provide an independent implementation in a different computer algebra system, 3) and to create an easily accessible, user-friendly interface to methods that are relevant to many disciplines outside of computer algebra. Additionally, we implemented some very recent algorithms for creative telescoping [7, 2] and closure properties [5].

Due to space restrictions, we will limit this short exposition to the top-level functionalities of HolonomicFunctions, which are: 1) converting a mathematical expression into a holonomic description, 2) executing holonomic closure properties, and 3) creative telescoping for general holonomic functions. More details, also about lower-level functionalities, can be found in [6], and detailed descriptions of all available commands are listed in the user's guide [8]. These documents, the package itself, and a collection of examples can be downloaded from the website

http://www.risc.jku.at/research/combinat/software/HolonomicFunctions/

(the required password is given for free to researchers and non-commercial users). HolonomicFunctions has first been released in 2009 and since then it has been extended constantly.

In order to apply the holonomic systems approach to some special function identity, the first step consists in converting all input expressions to holonomic descriptions (strictly speaking, our package works with descriptions of *∂-finite functions* [3], but for sake of simplicity, we don't want to insist on this subtle difference for now). This means, for a given mathematical function f, determine all linear partial (q-) difference / differential equations with polynomial coefficients that f satisfies ("holonomic system"). This usually infinite set of equations has the structure of a left ideal in the corresponding operator algebra; we use *Ore algebras* for representing such equations. The command to obtain an *annihilating ideal* (ideal of annihilating operators) for f is **Annihilator**; it takes as input a mathematical expression f, together with a list of operator symbols like D_x, S_n, etc. to specify the type of equations, and outputs the reduced left Gröbner basis of an annihilating ideal for f (note, however, that it is not guaranteed to be the maximal one so that it may not contain all annihilating operators for f). Following [4], we have extended our package such that it can also deal with certain non-holonomic functions. The **Annihilator** command recognizes more than 100 elementary and special functions (holonomic and non-holonomic) that are available in the Mathematica system.

The class of holonomic functions exhibits some nice and useful closure properties, among them addition, multiplication, certain substitutions and application of operators: given annihilating ideals of functions f and g, respectively, there are algorithms for computing an annihilating ideal of $f+g$, $f \cdot g$, etc. Since these operations are implemented on the level of ∂-finite functions, the corresponding commands are **DFinitePlus**, **DFiniteTimes**, **DFiniteSubstitute**, and **DFiniteOreAction**. We refer to [6] where these closure properties are explained in detail. We want to emphasize that the **Annihilator** command analyzes the syntactical structure of an input expression and then makes use of the above closure properties, e.g., multiplication in the input line In[2], see below.

It is a classic result that holonomic functions are also closed under taking integrals and sums; for these operations, the method of *creative telescoping* is employed. The problem of computing creative telescoping relations (consisting of the *telescoper* and the *certificate*) has attracted a great deal of attention during the last years. Zeilberger's solution in [11] (later coined "the slow algorithm") is based on elimination; in HolonomicFunctions it can be achieved via the **OreGroebnerBasis** command or the **FindRelation** command. The latter finds an element in a given left ideal that satisfies certain properties, to be specified by the user. Takayama's algorithm [9] is also based on elimination, but is more efficient than the slow algorithm; the command **Takayama** exports it to the user. The command **CreativeTelescoping** computes telescopers and certificates using Chyzak's algorithm [3], whereas the command **FindCreativeTelescoping** executes the heuristic fast approach proposed in [7]. Very recently, a creative telescoping algorithm for bivariate hyperexponential functions based on Hermite reduction [2] has been implemented by the author; it will be available in the next release of HolonomicFunctions.

At the end, we come back to Zeilberger's seminal paper [11] and demonstrate the usage of our package on the toy example that he discusses in the introduction, namely the generating function of the venerable Legendre polynomials:

$$\sum_{n=0}^{\infty} P_n(x)t^n = (1 - 2xt + t^2)^{-1/2}.$$

We start by computing an annihilating ideal for the summand on the left-hand side, and perform creative telescoping on it; this gives a list of telescopers and corresponding certificates:

In[1]:= **<< HolonomicFunctions.m**
HolonomicFunctions package by Christoph Koutschan, RISC-Linz, Version 1.6 (12.04.2012)

In[2]:= **ann = Annihilator[LegendreP[n, x]*t^n, {S[n], Der[t], Der[x]}]**

Out[2]= $\{tD_t - n, (n+1)S_n + (t - tx^2)D_x + (-ntx - tx), (x^2 - 1)D_x^2 + 2xD_x + (-n^2 - n)\}$

In[3]:= **{ts, cs} = CreativeTelescoping[ann, S[n] − 1, {Der[t], Der[x]}]**

Out[3]= $\left\{\left\{(-t^2 + 2tx - 1)D_x + t,\ (t^2 - 2tx + 1)D_t + (t - x)\right\}, \left\{(tx - 1)D_x - nt,\ (x-1)(x+1)D_x + \frac{n - ntx}{t}\right\}\right\}$

The correctness of this result can be verified by showing that the creative telescoping operators are indeed in the annihilating ideal (done here only for the first one, note the usage of noncommutative arithmetic):

In[4]:= **ct1 = ts[[1]] + (S[n] − 1) ** cs[[1]]**

Out[4]= $(tx - 1)S_n D_x - (n+1)tS_n + (tx - t^2)D_x + (nt + t)$

In[5]:= **OreReduce[ct1, ann]**

Out[5]= 0

Finally, one computes an annihilating ideal for the right-hand side: it agrees with the left ideal generated by the two telescopers above.

In[6]:= **Annihilator[(1 − 2*x*t + t^2)^(−1/2), {Der[t], Der[x]}]**

Out[6]= $\{(t^2 - 2tx + 1)D_x - t,\ (t^2 - 2tx + 1)D_t + (t - x)\}$

Comparing initial values (not done here) completes the proof.

References

[1] Gert Almkvist and Doron Zeilberger. The method of differentiating under the integral sign. *Journal of Symbolic Computation*, 10(6):571–591, 1990.

[2] Alin Bostan, Shaoshi Chen, Frédéric Chyzak, Ziming Li, and Guoce Xin. Hermite reduction and creative telescoping for hyperexponential functions. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC)*, New York, NY, USA, 2013. ACM. To appear (preprint on arXiv:1301.5038).

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

[4] Frédéric Chyzak, Manuel Kauers, and Bruno Salvy. A non-holonomic systems approach to special function identities. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 111–118, New York, NY, USA, 2009. ACM.

[5] Stavros Garoufalidis and Christoph Koutschan. Twisting q-holonomic sequences by complex roots of unity. In Joris van der Hoeven and Mark van Hoeij, editors, *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 179–186. ACM, 2012.

[6] Christoph Koutschan. *Advanced applications of the holonomic systems approach.* PhD thesis, Research Institute for Symbolic Computation (RISC), Johannes Kepler University, Linz, Austria, 2009.

[7] Christoph Koutschan. A fast approach to creative telescoping. *Mathematics in Computer Science*, 4(2-3):259–266, 2010.

[8] Christoph Koutschan. HolonomicFunctions (user's guide). Technical Report 10-01, RISC Report Series, Johannes Kepler University, Linz, Austria, 2010. http://www.risc.jku.at/research/combinat/software/HolonomicFunctions/.

[9] Nobuki Takayama. An algorithm of constructing the integral of a module—an infinite dimensional analog of Gröbner basis. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 206–211, New York, NY, USA, 1990. ACM.

[10] Doron Zeiberger. A fast algorithm for proving terminating hypergeometric identities. *Discrete Mathematics*, 80(2):207–211, 1990.

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