Koepf, W.

Algorithmische Bestimmung von Differentialgleichungen

Eine Funktion, die eine homogene, lineare Differentialgleichung mit polynomialen Koeffizienten erfüllt, nennen wir *einfach*. Beispiele einfacher Funktionen sind die rationalen und die algebraischen Funktionen, Potenz-, Exponential- und die Logarithmusfunktion, die Sinus- und die Kosinusfunktion, die inverse Sinus- und die inverse Tangensfunktion, ferner viele spezielle Funktionen wie Besselfunktionen und orthogonale Polynome. Dagegen ist die Tangensfunktion nicht einfach.

Stanley [4] bewies 1980 mit algebraischen Methoden, daß die Summe, das Produkt, ferner Ableitung und Stammfunktion einfacher Funktionen sowie die Komposition einfacher Funktionen mit rationalen Funktionen und rationalen Potenzen ebenfalls einfache Funktionen sind. Damit sieht man einer großen Klasse von Funktionen unmittelbar die Existenz einer einfachen Differentialgleichung an.

In [2] wurden Algorithmen für diese Fragestellungen veröffentlicht, die ich in dem Computeralgebrasystem Mathematica implementierte. Ähnliche Beobachtungen stammen von Zeilberger [5], und Salvy und Zimmermann haben eine Implementierung in Maple vorgenommen [3].

Sind f und g einfach der Ordnung n bzw. m, dann bestehen die Algorithmen für Summe und Produkt im wesentlichen in iterativer Differentiation der gegebenen Differentialgleichungen und der Anwendung des Gaußschen Algorithmus' zur Elimination der Funktionen $f^{(l)}$ $(l = 0, \ldots, n-1)$ und $g^{(l)}$ $(l = 0, \ldots, m-1)$ für die Summe bzw. $f^{(j)}\, g^{(k)}$ $(j = 0, \ldots, n-1,\ k = 0, \ldots, m-1)$ für das Produkt.

Daher ist $f + g$ einfach der Ordnung $\leq n + m$ und $f\,g$ ist einfach der Ordnung $\leq nm$. Weiter ist $f \circ r$ einfach der Ordnung n, falls r rational ist, und der Ordnung $\leq nq$ falls $r(x) = x^{p/q}$ $(p, q \in \mathbb{Z})$. In [2] wurde durch Beispiele gezeigt, daß alle Schranken scharf sind.

Hier sind einige Ergebnisse der Mathematica-Implementierung:

```
In[1]:= SumDE[f''[x]+f[x]==0,f'[x]-f[x]==0,f,x]

                                   (3)
Out[1]= f[x] - f'[x] + f''[x] - f     [x] == 0

In[2]:= ProductDE[f''[x]+f[x]==0,f'[x]-f[x]==0,f,x]

Out[2]= 2 f[x] - 2 f'[x] + f''[x] == 0

In[3]:= SumDE[f''[x]+x*f[x]==0,f'[x]-f[x]==0,f,x]

                     2                                                        (3)
Out[3]= (-1 + x + x ) f[x] - x (1 + x) f'[x] + (2 + x) f''[x] + (-1 - x) f    [x] == 0

In[4]:= ProductDE[f''[x]+x*f[x]==0,f'[x]-f[x]==0,f,x]

Out[4]= (1 + x) f[x] - 2 f'[x] + f''[x] == 0
```

Man stellt fest, daß die Methode unabhängig von der „Kompliziertheit" der betrachteten Funktionen ist. Während im ersten Beispiel die Exponentialfunktion mit einer trigonometrischen Funktion verknüpft wird, wird im zweiten Beispiel die Exponentialfunktion mit einer Airy-Funktion verknüpft: Die die (möglicherweise komplizierten) Funktionen darstellenden Objekte sind die viel einfacheren Differentialgleichungen.

Entsprechend nennen wir eine Funktion f_n einer diskreten Variablen n einfach der Ordnung m, wenn sie einer homogenen, linearen Rekursionsgleichung mit polynomialen Koeffizienten genügt.

Die obigen Algorithmen können ohne Umschweife auf diesen Fall übertragen werden, um Rekursionsgleichungen für $f_n + g_n$ und $f_n g_n$ zu berechnen. Hier sind einige Beispiele der MATHEMATICA-Implementierung:

```
In[5]:= SumRE[a[n+1]==a[n]/(n+1),a[n+1]==a[n],a,n]

                                   2
Out[5]= (1 + n) a[n] + (-1 - 3 n - n ) a[1 + n] + n (2 + n) a[2 + n] == 0

In[6]:= ProductRE[a[n+1]==a[n]/(n+1),a[n+1]==a[n],a,n]

Out[6]= a[n] + (-1 - n) a[1 + n] == 0

In[7]:= SumRE[a[n+1]==a[n]/(n+1),a[n+1]==(n+1)*a[n],a,n]

               2                           2              2                       2
Out[7]= (1 + n)  (3 + n) a[n] - (1 + 3 n + n ) (3 + 3 n + n ) a[1 + n] + n (2 + n)  a[2 + n] == 0

In[8]:= ProductRE[a[n+1]==a[n]/(n+1),a[n+1]==(n+1)*a[n],a,n]

Out[8]= a[n] - a[1 + n] == 0
```

Die vorgestellten Algorithmen eignen sich zur Berechnung von Differentialgleichungen spezieller Funktionen und damit zur Verifikation von Identitäten. Um z. B. die Identität

$$L_n^{(-1/2)}(x) = \frac{(-1)^n}{n!\,2^{2n}}\,H_{2n}\left(\sqrt{x}\right)$$

zwischen den verallgemeinerten Laguerre- und Hermite-Polynomen zu beweisen, läßt man sich die gemeinsame Differentialgleichung

```
In[9]:= SimpleDE[LaguerreL[n,-1/2,x],x]

Out[9]= 2 n F[x] + (1 - 2 x) F'[x] + 2 x F''[x] == 0

In[10]:= SimpleDE[HermiteH[2n,Sqrt[x]],x]

Out[10]= 2 n F[x] + (1 - 2 x) F'[x] + 2 x F''[x] == 0
```

der rechten und linken Seite berechnen und verifiziert die Anfangsbedingungen.

Dieser Verifikationsalgorithmus läßt sich auf den Fall von Rodriguesformeln wie z. B.

$$(-1)^n\,2^n\,n!\,e^{x^2}\;\mathrm{erfc}_n(x) = \frac{\partial^n}{\partial x^n}\left(e^{x^2}\,\mathrm{erfc}\,x\right)$$

sowie Summenformeln wie

$$\sum_{k=0}^{n} L_k^{(\alpha)}(x) = L_n^{(\alpha+1)}(x)\ .$$

ausdehnen [1].

1. Literatur

1 KOEPF, W.: Algorithmic work with orthogonal polynomials and special functions. Konrad-Zuse-Zentrum Berlin (ZIB), Preprint SC 94-5, 1994.

2 KOEPF, W.; SCHMERSAU, D.: Spaces of functions satisfying simple differential equations. Konrad-Zuse-Zentrum Berlin (ZIB), Technical Report TR 94-2, 1994.

3 SALVY, B.; ZIMMERMANN, P.: GFUN: A package for the manipulation of generating and holonomic functions in one variable. Rapports Techniques **143**, INRIA, Rocquencourt, 1992.

4 STANLEY, R. P.: Differentiably finite power series. Europ. J. Combinatorics **1** (1980), 175–188.

5 ZEILBERGER, D.: A holonomic systems approach to special functions identities. J. Comput. Appl. Math. **32** (1990), 321-368.

Anschrift: KOEPF, W., Konrad-Zuse-Zentrum für Informationstechnik Berlin, Heilbronner Str. 10, D-10711 Berlin.