

Extended Rate, more GFUN

Martin Rubey

Fakultät für Mathematik, Universität Wien

We present a software package that guesses formulas for sequences of, for example, rational numbers or rational functions, given the first few terms. Thereby we extend and complement Christian Krattenthaler's program `Rate` and the relevant parts of Bruno Salvy and Paul Zimmermann's `GFUN`.

1 Introduction

For some a brain-teaser, for others one step in proving their next theorem: given the first few terms of a sequence of, say, integers, what is the next term, what is the general formula? Of course, no unique solution exists, however, by Occam's razor, we will prefer a 'simple' formula over a more 'complicated' one. In this article we present a new package `Guess` written for the computer algebra system `Axiom`. Both `Guess` and `Axiom` are freely available and can be tried out online at <http://wiki.axiom-developer.org/GuessingFormulas>.

Some sequences are very easy to 'guess', like

$$1, 4, 9, 16, \dots, \quad (1)$$

or

$$1, 1, 2, 3, 5, \dots \quad (2)$$

Others are a little harder, for example

$$0, 1, 3, 9, 33, \dots \quad (3)$$

Of course, at times we might want to guess a formula for a sequence of polynomials, too:

$$1, 1 + q + q^2, (1 + q + q^2)(1 + q^2), (1 + q^2)(1 + q + q^2 + q^3 + q^4) \dots, \quad (4)$$

or

$$\begin{aligned} &1, 1, 1 + q, 1 + q + q^2, 1 + q + q^2 + q^3 + q^4, 1 + q + q^2 + q^3 + 2q^4 + q^5 + q^6, \\ &1 + q + q^2 + q^3 + 2q^4 + 2q^5 + 2q^6 + q^7 + q^8 + q^9, (1 + q^4 + q^6)(1 + q + q^2 + q^3 + q^4 + q^5 + q^6), \\ &(1 + q^4)(1 + q + q^2 + q^3 + q^4 + q^5 + 2q^6 + 2q^7 + 2q^8 + 2q^9 + q^{10} + q^{11} + q^{12}), \dots \end{aligned} \quad (5)$$

Fortunately, with the right tool, it is a matter of a moment to figure out formulas for all of these sequences. In this article we describe a computer program that encompasses well known techniques and adds new ideas that we hope to be very effective. In particular, we generalize both Christian Krattenthaler's program `Rate` [2], and the guessing functions present in `GFUN` written by Bruno Salvy and Paul Zimmermann [3]. In a future version we will also provide means to guess multivariate formulas, along the lines of Doron Zeilberger's programs `GuessRat` and `GuessHolo` [5].

Finally, we would also like to mention *The online encyclopedia of integer sequences* of Neil Sloane [4]. There, you can enter a sequence of integers and chances are good that the website will respond with one or more likely matches. However, the approach taken is quite different from ours: the encyclopedia keeps a list of currently 117,520 sequences, entered more or less manually, and it compares the given sequence with each one of those. Besides that, it tries some simple transformations on the given sequence to find a match. Furthermore it tries some simple programs we will describe below to find a formula, although with a time limit, i.e., it gives up when too much time has elapsed.

Thus, the two approaches complement each other: For example, there are sequences where no simple formula is likely to exist, and which can thus be found only in the encyclopedia. On the other hand, there are many sequences that have not yet found their way into the encyclopedia, but can be guessed in a few minutes by your computer.

2 Function Classes Suitable for Guessing

In this section we briefly present the function classes which are covered by our package. Note that interpolation in various flavours is at the heart of all algorithms used in our package. In most circumstances it turns out the the very general fraction free method recently given by Bernhard Beckermann and George Labahn [1] is the fastest.

Throughout this section, $n \mapsto f(n)$ is the function we would like to guess, and $F(z) = \sum_{n \geq 0} f(n)z^n$ its generating function.

Rational functions - `guessRat`

A function $n \mapsto f(n)$ is rational, if it is the quotient of two polynomials, i.e., $f(n) = \frac{p(n)}{q(n)}$.

Given $f(0), f(1), \dots, f(m)$, we can use rational interpolation to reconstruct f . Since we do not know the degrees of the polynomials p and q , we determine the interpolating functions for all combinations of degrees, such that the total degree equals $m - 1$. This is also the method used by Christian Krattenthaler's program `Rate` [2].

Rational functions with an Abelian term - `guessExpRat`

This is the class of functions $f(n) = (a + bn)^n \frac{p(n)}{q(n)}$, where p and q are polynomials. Such functions arise, for example, in tree enumeration. It seems that our package is the first to be able to guess such functions. A detailed description of the algorithm will appear elsewhere.

Recurrences with constant coefficients - `guessPade`

That is, functions $f(n)$ satisfying

$$c_0 f(n) + c_1 f(n-1) + c_2 f(n-2) + \dots + c_k f(n-k) = 0.$$

Note that these sequences can also be defined as coefficients of rational generating functions, i.e.,

$$\sum_{n \geq 0} f(n)z^n = \frac{p(z)}{q(z)},$$

where p and q are polynomials.

Similar to interpolation, we can use Padé approximation to recover p and q from $f(0), f(1), \dots, f(m)$. Again we determine the approximants for all combinations of degrees of p and q . This method is also provided by `GFUN`.

Recurrences with polynomial coefficients - `guessPRec`

The maybe most important class of functions, at least in combinatorics, consists of those $f(n)$ that satisfy

$$c_0(n)f(n) + c_1(n)f(n-1) + c_2(n)f(n-2) + \dots + c_k(n)f(n-k) = 0,$$

where the c_i are polynomials for $i \in \{0, 1, \dots, k\}$. The *degree* of the recurrence is defined to be the maximal degree of the c_i , while its *order* is k .

Equivalently, these sequences are coefficients of holonomic generating functions, i.e., functions $F(z) = \sum_{n \geq 0} f(n)z^n$ that satisfy

$$c_0(z)F(z) + c_1(z)F'(z) + c_2(z)F''(z) + \dots + c_k(z)F^{(k)}(z) = 0,$$

where the c_i are polynomials. It can be shown that this class is a superset of the class of algebraic functions, i.e., functions $F(z)$ that satisfy an equation of the form

$$d_1(z)F(z) + d_2(z)F^2(z) + \dots + d_k(z)F^k(z) = 0,$$

where the d_i are again polynomials.

To obtain a recurrence from $f(0), f(1), \dots, f(m)$, one can, for all combinations of degree d and order o such that $(d+1) * (o+1) = m-1$, set up a system of linear equations, which is then solved by standard methods. This is the approach taken by `GFUN`.

However, it turns out that it is more efficient to use the interpolation algorithm by Bernhard Beckermann and George Labahn [1].

Differentially algebraic series - guessADE

A function $F(z)$ is *differentially algebraic* if it satisfies an *algebraic differential equation*, i.e., an equation of the form

$$p\left(F(z), F'(z), F''(z), \dots, F^{(k)}(z)\right) = 0,$$

where p is a polynomial. Clearly, every holonomic function is also differentially algebraic.

This class of functions has received interest in combinatorics only very recently, although it comprises several important generating functions. For example, the (exponential) generating function for the Bell numbers B_n , counting the number of partitions of $\{1, 2, \dots, n\}$, is

$$B(z) = \sum_{n \geq 0} B_n \frac{z^n}{n!} = e^{e^z - 1}.$$

This function is not holonomic, but it satisfies the simple algebraic differential equation $B''B - (B')^2 - B'B = 0$.

It appears that our package is the first to be able to guess such equations.

3 Operators

We stated in the previous section, that the method used by `Rate` would be rational interpolation. Of course, this is a half-lie, since otherwise the program would be rather useless.

The truth is that it occurs frequently that although a formula is not rational, applying one of the two following operators makes it rational:

Δ_n the differencing operator, transforming $f(n)$ into $f(n) - f(n-1)$, and

Q_n the operator that transforms $f(n)$ into $f(n)/f(n-1)$.

Thus, we can simply apply these operators recursively to the sequence, until we are able to guess a formula using interpolation, Padé approximation or one of its extensions. Note that `Rate` applies only the operator Q_n , while `GFUN` applies neither of them.

4 Some remarks

- All of the presented guessing algorithms are insensitive to the shift operator. I.e., if a formula for the sequence $(s_1, s_2, \dots, s_m)$ can be guessed, then the corresponding formula for $(s_2, s_3, \dots, s_{m+1})$ will be found, too.
- All classes of functions presented in Section 2 are closed under multiplication. I.e., if formulas for $(s_1, s_2, \dots, s_m)$ and $(t_1, t_2, \dots, t_m)$ can be guessed, then also for $(s_1 t_1, s_2 t_2, \dots, s_m t_m)$. Furthermore, all classes except the one covered by `guessExpRat`, all are closed under addition. However, the class of functions covered by `guessExpRat` is not even closed under addition of a constant! It would be good to have a remedy for this, i.e., a more general class of functions, which is still suitable for guessing.

References

- [1] Bernhard Beckermann and George Labahn, *Fraction-free computation of matrix rational interpolants and matrix GCDs*, SIAM Journal on Matrix Analysis and Applications **22** (2000), no. 1, 114–144 (electronic). MR MR1779720 (2001e:65024)
- [2] Christian Krattenthaler, *RATE: A Mathematica guessing machine*, <http://mat.univie.ac.at/~kratt/rate/rate.html>.
- [3] Bruno Salvy and Paul Zimmerman, *GFUN: a maple package for the manipulation of generating and holonomic functions in one variable*, Transactions on Mathematical Software **20** (1994), no. 2, 163–177, <http://pauillac.inria.fr/algo/libraries>.
- [4] N. J. A. Sloane, *The on-line encyclopedia of integer sequences*, Notices of the American Mathematical Society **50** (2003), no. 8, 912–915, <http://www.research.att.com/~njas/sequences>. MR MR1992789 (2004f:11151)
- [5] Doron Zeilberger, *The HOLONOMIC ANSATZ I. Foundations and Applications to Lattice Path Counting*.

