IntegerSequences: a package for computing with k-regular sequences

Eric Rowland

Department of Mathematics
Hofstra University
Hempstead, NY 11549, USA
eric.rowland@hofstra.edu

Abstract. IntegerSequences is a *Mathematica* package for computing with integer sequences. Its support for k-regular sequences includes basic closure properties, guessing recurrences, and computing automata. Recent applications have included establishing the structure of extremal a/b-power-free words, obtaining a product formula for the generating function enumerating binomial coefficients by their p-adic valuations, and proving congruences for combinatorial sequences modulo prime powers.

Keywords: integer sequences, regular sequences, automatic sequences

1 Introduction

IntegerSequences [9] is a *Mathematica* package for identifying and computing with integer sequences from a variety of classes. It has a particular emphasis on the class of k-regular sequences, which arise widely in combinatorics, number theory, and theoretical computer science. The following code loads the package, assuming it is downloaded to one of the directories listed in `$Path` (the recommended location being the Applications subdirectory of `$UserBaseDirectory`).

```
In[1]:= << IntegerSequences`
```

A notebook version of this extended abstract containing executable code is available from the author's web site[1].

The following set of subsequences is central to the definition of a k-regular sequence.

Definition 1. *Let $k \geq 2$ be an integer. The* k-kernel *of a sequence $s(n)_{n \geq 0}$ is the set*

$$\{s(k^e n + i)_{n \geq 0} : e \geq 0 \text{ and } 0 \leq i \leq k^e - 1\}.$$

The k-kernel is the base-k analogue of the set of shifts $\{s(n+i)_{n \geq 0} : i \geq 0\}$. A sequence $s(n)_{n \geq 0}$ (such as the Fibonacci sequence) is *constant-recursive* if $\{s(n+i)_{n \geq 0} : i \geq 0\}$ is contained in a finite-dimensional vector space. We define k-regular (or k-constant-recursive) sequences analogously.

[1] https://wolfr.am/uZ4DJDth

Definition 2. *Let $k \geq 2$ be an integer. A sequence $s(n)_{n\geq 0}$ with entries in a field F is* k-regular *if its k-kernel is contained in a finite-dimensional F-vector space.*

For example, consider the *ruler sequence* [6, A007814]

$$0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0, 4, \ldots$$

whose nth term $s(n)$ is the exponent of 2 in the prime factorization of $n + 1$. The ruler sequence is 2-regular, since the recurrence

$$\begin{aligned} s(2n) &= 0 \\ s(4n+1) &= -s(n) + s(2n+1) \\ s(4n+3) &= -s(n) + 2s(2n+1) \end{aligned} \tag{1}$$

establishes that the 2-kernel is contained in the $\mathbb{Q}$-vector space generated by $s(n)_{n\geq 0}$ and $s(2n+1)_{n\geq 0}$.

The class of k-regular sequences was introduced by Allouche and Shallit [1], who established several equivalent characterizations and a number of fundamental properties. In particular, $s(n)_{n\geq 0}$ is k-regular if and only if there exists some integer $r \geq 0$ (the dimension of the associated vector space), $r \times r$ matrices $M(0), M(1), \ldots, M(k-1)$, a $1 \times r$ vector u, and an $r \times 1$ vector v such that

$$s(n) = u\, M(n_0)\, M(n_1) \cdots M(n_\ell)\, v$$

for all $n \geq 0$, where $n_\ell \cdots n_1 n_0$ is the standard base-k representation of n [1, Lemma 4.1]. For example, the ruler sequence can be represented by

$$u = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad M(0) = \begin{bmatrix} 0 & 0 \\ -1 & 1 \end{bmatrix} \qquad M(1) = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix} \qquad v = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The matrices $M(0)$ and $M(1)$ encode the recurrence (1). The vector v contains the 0th term of each generator sequence, namely $s(0) = 0$ and $s(2 \cdot 0 + 1) = s(1) = 1$. The vector u specifies which linear combination of the generators we are interested in, namely $s(n)_{n\geq 0} = 1 \cdot s(n)_{n\geq 0} + 0 \cdot s(2n+1)_{n\geq 0}$.

IntegerSequences uses the matrices $M(d)$ and the vectors u, v to represent a k-regular sequence. The syntax is as follows.

```
In[2]:= s = RegularSequence[{1, 0}, {{{0, 0}, {-1, 1}}, {{0, 1}, {-1, 2}}}, {0, 1}];
```

The design of `RegularSequence` parallels the built-in *Mathematica* symbol for representing a holonomic sequence, `DifferenceRoot`[2]. Passing an argument to a `RegularSequence` object computes a term of the sequence.

```
In[3]:= Table[s[n], {n, 0, 15}]
Out[3]= {0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0, 4}
```

```
In[4]:= Clear[s]
```

[2] `http://reference.wolfram.com/language/ref/DifferenceRoot.html`

Basic closure properties for k-regular sequences established in [1] are implemented in the function `RegularSequenceReduce`, which attempts to reduce an expression to a single `RegularSequence` object. The following writes the Stern–Brocot and Thue–Morse sequences as 2-regular sequences and then computes their sum.

```
In[5]:= RegularSequenceReduce[SternBrocot[n] + ThueMorse[n], n] //
          RegularSequenceMatrixForm
```

Out[5]= $\texttt{RegularSequence}\left[\{1, 0, 1, 0\}, \left\{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 2 \end{pmatrix}\right\}, \{0, 1, 0, 1\}, 2\right][n]$

2 Guessing a k-regular sequence

Given the first N terms of a sequence, one is frequently interested in guessing a general form for the sequence. A procedure for guessing k-regular sequences was described by Shallit [14]. The implementation in INTEGERSEQUENCES works by maintaining a set B of generators, a set R of relations, and a set of k-kernel sequences S which have not yet been written as a linear combination of elements of B. Initialize $B = \{\}$, $R = \{\}$, and $S = \{s(n)_{n\geq 0}\}$. While $S \neq \{\}$, remove a sequence $t(n)_{n\geq 0}$ from S and determine, using the known terms, whether it is a linear combination of elements of B; if it is, add the linear relation to R; if it is not, add $t(n)_{n\geq 0}$ to B as a new generator and add its k subdivisions $t(kn + 0)_{n\geq 0}, \ldots, t(kn + (k-1))_{n\geq 0}$ to S. When S becomes empty, we have determined a conjectural basis B such that every element of the k-kernel can be written as a linear combination of the elements of B. The set of relations R, along with the initial term of each element of B, uniquely determines a sequence that agrees with $s(n)_{n\geq 0}$ on the known terms.

Since only finitely many terms of $s(n)_{n\geq 0}$ are known, it is possible that as we consider additional sequences from the k-kernel we will exhaust the known terms. If elements of B which were previously known to be linearly independent become linearly dependent due to truncating terms, then we do not have enough terms to confidently guess a recurrence.

This algorithm is implemented in `FindRegularSequenceFunction`. To our knowledge, this is the only publicly available guesser for k-regular sequences. The first argument is the list of terms, and the second argument is k. The following guesses a 2-regular representation for the number of 1s in the binary representation of n.

```
In[6]:= terms = Table[DigitCount[n, 2, 1], {n, 0, 100}];
        FindRegularSequenceFunction[terms, 2] // RegularSequenceMatrixForm
```

Out[7]= $\texttt{RegularSequence}\left[\{1, 0\}, \left\{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & 2 \end{pmatrix}\right\}, \{0, 1\}, 2\right]$

A variant, `FindRegularSequenceRecurrence`, uses the same algorithm but outputs a recurrence rather than a `RegularSequence` object.

```
In[8]:= Column[FindRegularSequenceRecurrence[terms, 2, s[n]]]

        s[2 n] == s[n]
        s[1 + 4 n] == s[1 + 2 n]
Out[8]= s[3 + 4 n] == -s[n] + 2 s[1 + 2 n]
        s[0] == 0
        s[1] == 1
```

More generally, FindRegularSequenceFunction supports guessing multidimensional $(k_1, \dots, k_d)$-regular sequences $s(n_1, \dots, n_d)_{n_1 \geq 0, \dots, n_d \geq 0}$. Let $\nu_p(n)$ be the p-adic valuation of n, that is, the exponent of the highest power of p dividing n. The 2-dimensional sequence consisting of the 2-adic valuations of $\binom{n}{m}$ (where we treat ∞ as a formal symbol) is 2-regular:

```
In[9]:= array = Table[IntegerExponent[Binomial[n, m], 2], {n, 0, 100}, {m, 0, 100}];
        FindRegularSequenceFunction[array /. ∞ → infinity, 2] //
         RegularSequenceMatrixForm
```

Out[10]= $$\text{RegularSequence}\left[\{1, 0, 0, 0\}, \left\{\left\{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}\right\}, \left\{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\right\}\right\}, \{0, \text{infinity}, \text{infinity}, 1\}, \{2, 2\}\right]$$

We mention several applications where FindRegularSequenceFunction has been used to guess a sequence or family of sequences.

For a rational number $\frac{a}{b} > 1$, a word w is an $\frac{a}{b}$*-power* if it can be written $v^e x$ where e is a non-negative integer, x is a prefix of v, and $\frac{|w|}{|v|} = \frac{a}{b}$. For example, 011101 is the $\frac{3}{2}$-power $(0111)^{3/2}$ with $v = 0111$ and $x = 01$. The lexicographically least $\frac{3}{2}$-power-free infinite word on the non-negative integers [6, A269518] is

$$0011021001120011031001130011021 0 \cdots .$$

It is difficult *a priori* to guess whether the sequence of letters in such a word is k-regular and, if so, to guess the correct value of k. However, through experimentation, FindRegularSequenceFunction revealed that the letters in the lexicographically least $\frac{3}{2}$-power-free word form a 6-regular sequence [11].

```
In[11]:= FindRegularSequenceFunction[
          {0, 0, 1, 1, 0, 2, 1, 0, 0, 1, 1, 2, 0, 0, 1, 1, 0, 3, 1, 0, 0, 1, 1,
           3, 0, 0, 1, 1, 0, 2, 1, 0, 0, 1, 1, 4, 0, 0, 1, 1, 0, 3, 1, 0, 0, 1,
           1, 2, 0, 0, 1, 1, 0, 2, 1, 0, 0, 1, 1, 3, 0, 0, 1, 1, 0, 3, 1, 0, 0, 1}, 6] //
         RegularSequenceMatrixForm
```

Out[11]= $$\text{RegularSequence}\left[\{1, 0, 0\}, \left\{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}\right\}, \{0, 0, 1\}, 6\right]$$

This discovery led to a large systematic study of the value of k for which the lexicographically least $\frac{a}{b}$-power-free word is k-regular, although it is an open question whether a k always exists [8].

Enumeration questions in combinatorics on words often turn out to have answers given by k-regular sequences for appropriate values of k. An explanation of this phenomenon in many cases was given by Charlier, Rampersad, and Shallit [2]. However, k-regular sequences also appear in enumeration questions not covered by their framework. For example, the ℓ-abelian complexity for many infinite words appears to be k-regular. The *ℓ-abelian complexity* of an infinite word counts factors up to ℓ-abelian equivalence — that is, two factors x and y are considered the same if $|x|_v = |y|_v$ for each word v of length $\leq \ell$. The 2-regularity of the 2-abelian complexities of two well-known words, the Thue–Morse word and the period-doubling word, were established by proving the 2-regularity of sequences satisfying a general reflection recurrence [7]. The 2-regular recurrences for such sequences were guessed by `FindRegularSequenceFunction`.

The intended use case of `FindRegularSequenceFunction` is a sequence of integers. However, the code is sufficiently general to support sequences of polynomials. Again, let $\nu_p(n)$ denote the p-adic valuation of n. Spiegelhofer and Wallner [15] considered the generating function counting binomial coefficients by their p-adic valuations $\nu_p(\binom{n}{m})$. For each prime p, `FindRegularSequenceFunction` is able to guess a p-regular recurrence for this generating function. For $p = 2$ we obtain the following. Note that the matrix entries are now polynomials in the formal variable x.

```
In[12]:= polynomials = Table[Sum_{m=0}^{n} x^IntegerExponent[Binomial[n,m],2], {n, 0, 100}];
         FindRegularSequenceFunction[polynomials, 2] // RegularSequenceMatrixForm
```

$$\text{Out[13]= RegularSequence}\left[\{1, 0\}, \left\{\begin{pmatrix} 0 & 1 \\ -2x & 1+2x \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 2 & x \end{pmatrix}\right\}, \{1, 1\}, 2\right]$$

The basis chosen by `FindRegularSequenceFunction` may not be the most natural basis, since it (a) necessarily consists of k-kernel elements and (b) depends on the order in which the k-kernel is traversed. By performing a suitable change of basis for each prime p, the author conjectured the following, which can be proved by a bijective argument [10].

Theorem 1. *Let p be a prime. For each $d \in \{0, 1, \ldots, p-1\}$, let $M_p(d)$ be the 2×2 matrix*

$$M_p(d) = \begin{bmatrix} d+1 & p-d-1 \\ d\,x & (p-d)\,x \end{bmatrix}.$$

Let $n \geq 0$, and let $n_\ell \cdots n_1 n_0$ be the standard base-p representation of n. Then

$$\sum_{m=0}^{n} x^{\nu_p(\binom{n}{m})} = \begin{bmatrix} 1 & 0 \end{bmatrix} M_p(n_0)\, M_p(n_1) \, \cdots \, M_p(n_\ell) \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

This theorem generalizes a well-known result of Fine [5] on the number of binomial coefficients not divisible by p. More generally, the analogous sequence of generating functions enumerating multinomial coefficients by their p-adic valuations is p-regular [10].

3 Computing automata for sequences modulo p^α

A k-regular sequence whose terms take on finitely many distinct values is called *k-automatic*. This name derives from the characterization of k-automatic sequences as sequences whose nth term is the output of an automaton when fed the base-k digits of n.

Many integer sequences that arise in combinatorics have the property that reducing each term modulo p^α produces a p-automatic sequence. For algebraic sequences modulo p, this is explained by Christol's theorem, which states that a sequence over a finite field of characteristic p is p-automatic if and only if its generating function is algebraic [3]. Therefore, if $\sum_{n\geq 0} s(n)x^n \in \mathbb{Z}[\![x]\!]$ is algebraic (as it is for the Catalan numbers, for example), then $\sum_{n\geq 0}(s(n) \bmod p)x^n \in \mathbb{F}_p[\![x]\!]$ is algebraic, so $(s(n) \bmod p)_{n\geq 0}$ is p-automatic. In IntegerSequences this is implemented in `AutomaticSequenceReduce`. The following computes a 3-automatic sequence, represented by an automaton, for the nth Catalan number modulo 3.

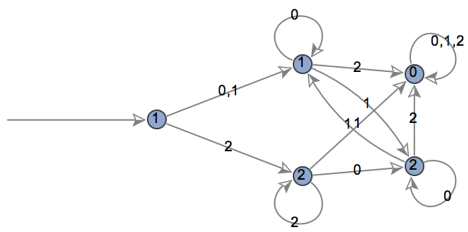
```
In[14]:= AutomaticSequenceReduce[Mod[CatalanNumber[n], 3], n]

Out[14]= AutomaticSequence[
        Automaton[{{1 → 2, 0}, {1 → 2, 1}, {1 → 3, 2}, {2 → 2, 0}, {2 → 4, 1},
          {2 → 5, 2}, {3 → 4, 0}, {3 → 5, 1}, {3 → 3, 2}, {4 → 4, 0}, {4 → 2, 1},
          {4 → 5, 2}, {5 → 5, 0}, {5 → 5, 1}, {5 → 5, 2}}, 1,
         {1 → 1, 2 → 1, 3 → 2, 4 → 2, 5 → 0}, InputAlphabet → {0, 1, 2}]][n]
```

The function `AutomatonGraph` produces the `Graph` object corresponding to an automaton.

```
In[15]:= AutomatonGraph[AutomaticSequenceAutomaton[%]]
```

Out[15]=

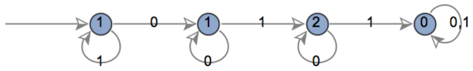


More generally, if $s(n)_{n\geq 0}$ is the diagonal of the power series of a multivariate rational expression $\frac{f}{g}$ whose denominator's constant term $g(0, \ldots, 0)$ is nonzero modulo p, then $(s(n) \bmod p^\alpha)_{n\geq 0}$ is p-automatic. An automaton for this sequence can be computed by embedding the p-kernel into the space of rational expressions with a certain fixed denominator [12]. The diagonal of a rational power series is represented in IntegerSequences by `DiagonalSequence`.

```
In[16]:= AutomaticSequenceReduce[Mod[DiagonalSequence[(1 - x)/(1 - (1 + x)^2 y), {x, y}][n], 4],
          n]
         AutomatonGraph[AutomaticSequenceAutomaton[%]]

Out[16]= AutomaticSequence[Automaton[{{1 → 2, 0}, {1 → 1, 1}, {2 → 2, 0},
            {2 → 3, 1}, {3 → 3, 0}, {3 → 4, 1}, {4 → 4, 0}, {4 → 4, 1}},
           1, {1 → 1, 2 → 1, 3 → 2, 4 → 0}, InputAlphabet → {0, 1}]][n]
```

Out[17]=



Computing automata for sequences modulo p^α provides routine proofs of many congruences that were established in the literature by nontrivial case analyses. For example, Eu, Liu, and Yeh [4] proved that no Motzkin number is divisible by 8. The following computation proves this in less than a second. The resulting automaton has 24 states.

```
In[18]:= automaton = AutomaticSequenceAutomaton[
            AutomaticSequenceReduce[Mod[MotzkinNumber[n], 8], n]];
         AutomatonStateCount[automaton]
         Sort[AutomatonOutputAlphabet[automaton]]

Out[19]= 24

Out[20]= {1, 2, 3, 4, 5, 6, 7}
```

Closely related to diagonal sequences are *constant-term sequences*. Let f and g be (possibly multivariate) Laurent polynomials, and let $s(n)$ be the constant term of $f^n g$. An automaton for $(s(n) \bmod p^\alpha)_{n\geq 0}$ can be computed similarly [13]. `AutomaticSequenceReduce` also implements this algorithm. Constant-term sequences are represented by `ConstantTermSequence`, where the first argument is f and the second argument is g.

```
In[21]:= automaton = AutomaticSequenceAutomaton[
            AutomaticSequenceReduce[
             Mod[ConstantTermSequence[1 + 1/x + x, 1 - x^2, x][n], 25], n]];
         AutomatonStateCount[automaton]
         Complement[Range[0, 24], AutomatonOutputAlphabet[automaton]]

Out[22]= 136

Out[23]= {0}
```

In fact that constant-term sequence is the sequence of Motzkin numbers, so we have established that no Motzkin number is divisible by 25.

For many sequences, including the sequences of Catalan and Motzkin numbers, the constant-term representation is preferable to the diagonal representation since it uses polynomials in a single variable, whereas the diagonal representation requires at least two variables.

References

1. Jean-Paul Allouche and Jeffrey Shallit, The ring of k-regular sequences, *Theoretical Computer Science* **98** (1992) 163–197.
2. Émilie Charlier, Narad Rampersad, and Jeffrey Shallit, Enumeration and decidable properties of automatic sequences, *International Journal of Foundations of Computer Science* **23** (2012) 1035–1066.
3. Gilles Christol, Teturo Kamae, Michel Mendès France, and Gérard Rauzy, Suites algébriques, automates et substitutions, *Bulletin de la Société Mathématique de France* **108** (1980) 401–419.
4. Sen-Peng Eu, Shu-Chung Liu, and Yeong-Nan Yeh, Catalan and Motzkin numbers modulo 4 and 8, *European Journal of Combinatorics* **29** (2008) 1449–1466.
5. Nathan Fine, Binomial coefficients modulo a prime, *The American Mathematical Monthly* **54** (1947) 589–592.
6. The On-Line Encyclopedia of Integer Sequences, `http://oeis.org`.
7. Aline Parreau, Michel Rigo, Eric Rowland, and Élise Vandomme, A new approach to the 2-regularity of the ℓ-abelian complexity of 2-automatic sequences, *The Electronic Journal of Combinatorics* **22** (2015) #P1.27.
8. Lara Pudwell and Eric Rowland, Avoiding fractional powers over the natural numbers, *The Electronic Journal of Combinatorics* **25** (2018) #P2.27.
9. Eric Rowland, IntegerSequences, `https://people.hofstra.edu/Eric_Rowland/packages.html#IntegerSequences`.
10. Eric Rowland, A matrix generalization of a theorem of Fine, *Integers* **18A** (2018) A18.
11. Eric Rowland and Jeffrey Shallit, Avoiding 3/2-powers over the natural numbers, *Discrete Mathematics* **312** (2012) 1282–1288.
12. Eric Rowland and Reem Yassawi, Automatic congruences for diagonals of rational functions, *Journal de Théorie des Nombres de Bordeaux* **27** (2015) 245–288.
13. Eric Rowland and Doron Zeilberger, A case study in meta-automation: automatic generation of congruence automata for combinatorial sequences, *Journal of Difference Equations and Applications* **20** (2014) 973–988.
14. Jeffrey Shallit, Remarks on inferring integer sequences, `https://cs.uwaterloo.ca/~shallit/Talks/infer.ps`.
15. Lukas Spiegelhofer and Michael Wallner, An explicit generating function arising in counting binomial coefficients divisible by powers of primes, *Acta Arithmetica* **181** (2017) 27–55.