Proceedings of the 12th International Conference on Quasicrystals (ICQ12)

Letter Frequencies in the Kolakoski Sequence

J. Nilsson
Fakultät für Mathematik, Universität Bielefeld, Postfach 100131, 33501 Bielefeld, Germany

The classical Kolakoski sequence is the unique sequence of two symbols {1, 2}, starting with 1, which is equal to the sequence of lengths of consecutive segments of the same symbol (run lengths). We discuss here numerical aspects of the calculation of the letter frequencies and how to find bounds for these frequencies.

DOI: 10.12693/APhysPolA.126.549
PACS: 02.70.Wz

1. Introduction

The classical Kolakoski sequence $K = (K_n)_{n=1}^{\infty}$ is the unique sequence on the alphabet $\{1, 2\}$ defined as the sequence of its own symbols' run lengths starting with a 1. The classical Kolakoski sequence was first studied in a work by Oldenburger [1], where it appears as the unique solution to the problem of a trajectory on the alphabet $\{1, 2\}$ which is identical to its exponent trajectory. The name of the Kolakoski sequence, however, originates from [2, 3]. In the *On-Line Encyclopedia of Integer Sequences* [4], the Kolakoski sequence has entry number A000002. The first letters of K are

```
K = 1   2     2    1 1    2    1 ...
    |  / \   / \   | |   / \   |           (1)
K = 1  2 2   1 1   2 1   2 2   1 ...
```

There are several interesting questions, answered and unanswered, on the properties of the classical Kolakoski sequence; Kimberling presents several of these in [5]. One of the simplest, and yet unresolved, question is that of the distribution of digits in K. If we let o_n be the number of 1s in K up to and including position n, that is $o_n = |\{i : K_i = 1, 1 \leq i \leq n\}|$, then the conjecture is

Conjecture. *The limit* $\lim_{n\to\infty} \frac{o_n}{n}$ *exists and is* $\frac{1}{2}$.

Both parts of the conjecture, the existence and the value, are still open. Several aspects of the conjecture (along with other properties and questions regarding the Kolakoski sequence) are considered by Dekking in [6–8]; see also the survey by Sing [9] and further references therein.

The outline of this paper is that we first present a highly memory efficient algorithm for generating the Kolakoski sequences, and with its help we calculate o_n up to $n = 10^{13}$. Thereafter we consider two different methods for finding bounds of the fraction o_n/n, which hold for all n larger than some N.

2. Generating the Kolakoski sequence

The Kolakoski sequence K can be generated by an algorithm in amortised linear time that uses only a logarithmic amount of memory (linear and logarithmic in the number of symbols generated), as presented in [10], and which we discuss here. Recall that amortised linear time means that the average work to perform n operations is made in linear time.

2.1. The algorithm

The idea in the algorithm is that if we set out only to find o_n, we do not have to save the complete sequence up to position n when stepping through the sequence K. As in the intuitive way of generating K, we look back at a previous position to see which symbol run to append. However, this previous position is itself determined by a letter even further back, and so on. If we keep track only of these positions that we "look back at", we can drastically reduce the amount of space needed. (If however the space restriction is lifted, the calculation of the number o_n can be accelerated, as presented by Rao [11].)

To get a hint on how this "looking back" can be done, we take as a starting point a scheme, as in (1). We see that the upper row defines (or conversely, may be defined as) the run lengths of the symbols in the lower one. We expand this scheme by adding more rows above and connecting each symbol to the symbol in the row above that has (via run length) generated it. In this way, we obtain a tree structure, as illustrated in Fig. 1, where $K = 1K'$.

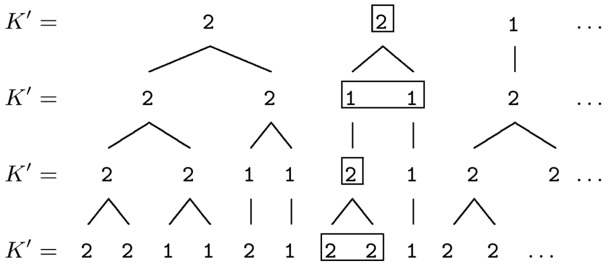

Fig. 1. Generating the Kolakoski sequence by looking back at already generated symbols. The boxed letters need to be stored. The scheme continues unboundedly upwards.

We may thus interpret the letters in the Kolakoski sequence K as the leaves of a tree (the leaves are the symbols in the bottom row in Fig. 1). Each internal node in this tree structure is a symbol in an upper row interpreted as a run length. Each letter is connected to the letter above that has generated it (called an ancestor), and also to the letter(s) below that it generates, termed children. This tree structure continues upwards without bound as we step through the symbols of K. However, we only need to go up in the tree until we find an ancestor, to the leaf we are currently looking at, at a left-most position.

The algorithm for finding o_n can concisely be described as an "in-order traverse" of this tree structure, where we start from the lower left, and where we keep track of the symbols we see in the leaves during the traversal. While traversing, we add new ancestors when needed; that is we build the tree as we traverse it. To reduce the memory requirement, we dynamically generate and keep track only of the part of the tree that we currently use for the traversal. While doing so, we store the ancestors along with an indicator that tells us which of its children we have already traversed. To this end, we introduce pointers P_k, which are assigned values from the set $S = \{1, 2, 11, 22\}$. Let us note that here a run is defined as a word from the set S. At any given time, the pointer P_0 holds the current run in the leaves and P_1 holds the ancestor to P_0. Similarly, any P_k that has been initiated holds the ancestor to P_{k-1}.

In [10], the following run time analysis result on the above algorithm is proved.

Proposition. *Let $P(n)$ be the number of pointers used by the algorithm to calculate o_n.*

1. *The amount of space used to find o_n is logarithmic in n. That is, $P(n) = O(\log n)$.*
2. *The algorithm runs in (amortized) linear time. That is, to find o_n we have to do an amount of work of order $O(n)$.*

The main ideas in the proof of the proposition are based on the use of the trivial bounds of the letters frequencies

$$\frac{1}{4} \le \frac{o_n}{t_n} \le 4$$

for $n \ge 2$, where $t_n = n - o_n$ is the number of 2s up to position n. This implies that if P_k holds the symbol at position n then P_{k+1} holds the symbol at at most position $\frac{5}{6}n$. By recursion, this implies a need for at most a logarithmic amount of space.

The run time result is based on a similar argument, plus the observation that we do not need to go higher in the tree for each increment of P_0. We only need to go higher when P_k contains a single symbol.

2.2. Calculations

An implementation and run of the above algorithm gives the result presented in Table I and in Fig. 2. The output indicates that the fraction o_n/n should tend to 1/2, when n grows, but clearly does not prove it. We denote for the Kolakoski sequence the maximal deviation of the proportion of 1s from $\frac{1}{2}$ in a logarithmic decade by

$$D(n) = \max_{\frac{1}{10}n < i \le n} \left| \frac{1}{2} - \frac{o_i}{i} \right|,$$

where o_i is the number of 1s up to position i.

TABLE I

The number of 1s in prefixes of the Kolakoski sequence. The column with the number of 1s is the sequence A195206 in the *On-Line Encyclopedia of Integer Sequences* [4].

n	Number of 1s	Pointers	$D(n)$
10^1	5	4	1.667×10^{-1}
10^2	49	10	8.333×10^{-2}
10^3	502	16	1.351×10^{-2}
10^4	4 996	22	3.588×10^{-3}
10^5	49 972	27	5.481×10^{-4}
10^6	499 986	33	2.800×10^{-4}
10^7	5 000 046	39	3.892×10^{-5}
10^8	50 000 675	44	2.054×10^{-5}
10^9	500 001 223	50	8.586×10^{-6}
10^{10}	4 999 997 671	56	2.152×10^{-6}
10^{11}	50 000 001 587	61	4.453×10^{-7}
10^{12}	500 000 050 701	67	2.140×10^{-7}
10^{13}	5 000 000 008 159	73	6.774×10^{-8}

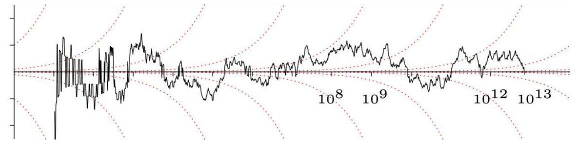

Fig. 2. A dynamic log–log plot around $\frac{1}{2}$ of the fraction of 1s in the Kolakoski sequence.

3. Bounds on the letter frequencies

The algorithm presented above only deals with the letter distribution in the beginning of the Kolakoski sequence K but does not say anything about the long time asymptotic behaviour of o_n/n. In the following sections, we present two approaches, based on the same idea, that give general bounds of the letter frequencies in K, without assuming that the limit in the conjecture exists.

Both methods presented here are based on the construction of a graph G, so that K is described as an infinite path in G. By constructing larger and larger graphs, we can extract more and more information about K, and thereby, with the help of exact computer enumerations, find bounds of the letter frequencies.

3.1. Induced graphs

We consider here an idea of Chvátal [12]. From the Kolakoski sequence K we construct a graph G_d in the following way; fix an integer $d \ge 1$ and write d copies of K under each other, so that the symbols in one row indicate the run length in the sequence above. This is

$$\begin{array}{c} d=3 \\ \text{copies} \end{array} \left\{ \begin{array}{lccccccccc|c|ccc|c|l} K= & 1 & 2 & 2 & 1 & 1 & 2 & 1 & 2 & 2 & 1 & 2 & 2 & 1 & 1 & 2\ldots \\ K= & 1 & & 2 & & 2 & 1 & 1 & & 2 & 1 & & 2 & & 2 & 1\ldots \\ K= & 1 & & & & 2 & & 2 & & 1 & 1 & & & & 2 & 1\ldots \end{array} \right.$$

Fig. 3. Three copies of the Kolakoski sequence that indicate how the graph G_3 is constructed.

illustrated in Fig. 3 for $d = 3$. In the scheme in Fig. 3, we look for columns of letters with maximal height, see the vertical boxes. These columns (or words) are taken as labels of the nodes in the induced graph G_3, compare Fig. 4. The arcs in G_3 are labelled with the sub-word of K in the top row of our scheme, which we see when going from one maximal column to the next.

The Kolakoski sequence is now given by an infinite path through G_d, for any fixed $d \geq 1$. An open and interesting question concerning the graphs G_d is whether they are strongly connected, that is whether there is a path from each vertex to every other vertex.

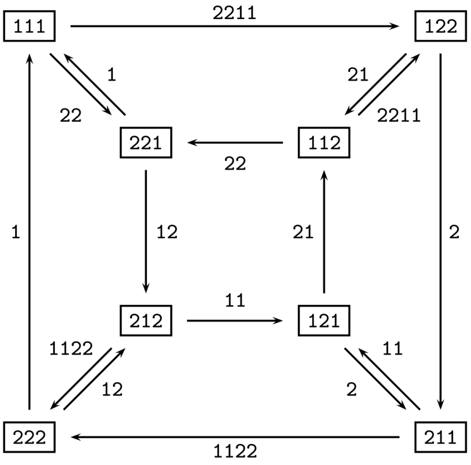

Fig. 4. The graph G_3 induced by the Kolakoski sequence. A cycle with highest number of 1s compared to the total number of symbols in the cycle is found in the lower right corner; the cycle over the nodes v_{121} and v_{211}.

3.2. Transducers

We consider here an idea based on transducers; see the web page by Rao [11]. The idea is based on the fact, as noted in [13], that one can obtain the Kolakoski sequence K by starting with 2 as a seed and iterate the two substitutions

$$\mu_0 : \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 11 \end{cases}, \qquad \mu_1 : \begin{cases} 1 \mapsto 2 \\ 2 \mapsto 22 \end{cases}$$

alternatively, i.e., μ_0 substitutes letters on even positions and μ_1 letters on odd positions

$$2 \mapsto 22 \mapsto 2211 \mapsto 221121 \mapsto 221121221 \mapsto \ldots$$

The alternations of substitutions can be described by the *finite state transducer* T_1 in Fig. 5. A finite state transducer (FST) is a finite state machine with two tapes: an input tape and an output tape. It works as follows: we are currently in one state and read on one of the tapes, then the transducer tells us, depending on what we read, what to write on the second tape and to which state we jump.

The sequence K is a fixed point to the transducer T_1, the other fixed point is $K = 1K'$. By composing the transducer T_1 with itself we obtain $T_2 = T_1 \circ T_1$, see Fig. 6. This easily generalises to higher level transducers $T_n = T_{n-1} \circ T_1$. The Kolakoski sequence is now, as for the first described type of graphs, given by an infinite path through the vertices of T_n, for any fixed $n \geq 1$.

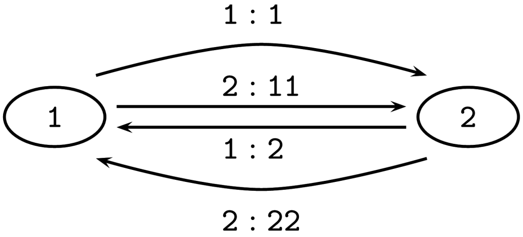

Fig. 5. The finite state transducer T_1 for the Kolakoski sequence.

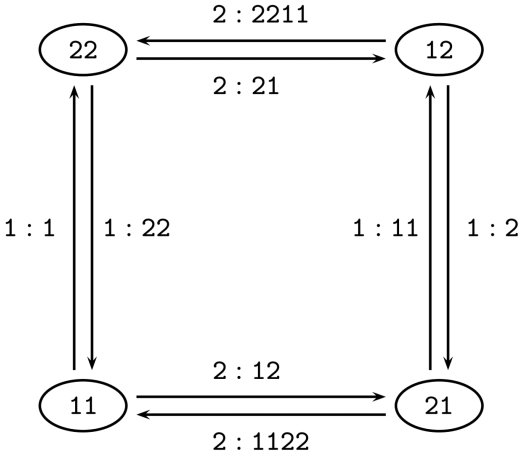

Fig. 6. The transducer T_2. A cycle with highest number of 1s compared to the total number of symbols in the cycle is found on right side, the cycle over the nodes v_{12} and v_{21}.

3.3. Calculations

The graph G_d is equivalent to the transducer T_{d-1}, as there is a graph morphism $\eta : V(G_d) \to V(T_{d-1})$. Therefore, it is enough to consider G_d. To obtain bounds for the letter frequencies in K we look for a cycle in G_d with highest ratio of 1s. This is since an infinite path in G_d describing K may in the worst case end in such a cycle. Such a cycle then gives a bound of the letter frequencies. (Note that no path describing K can end in a cycle, since this would imply that K ends with an infinite repetition of the same symbol.)

Let u_d be the largest ratio of 1s to all symbols of any cycle of the graph G_d (or in T_{d-1}), that is

$$u_d := \max_{c \text{ a cycle in } G_d} \frac{\text{number of 1s in } c}{\text{total number of letters in } c}.$$

In Table II, we present the value of u_d for small values of d, see also [11]. From the computer calculations, we obtain that there is an $N \geq 1$ such that

$$\sup_{n\geq N} \left| \frac{o_n}{n} - \frac{1}{2} \right| \leq \frac{455920839}{911696379} - \frac{1}{2} \leq 0.000080,$$

where o_n is the number of 1s among the first n letters in the Kolakoski sequence. Let us note here that we do not assume that the letter frequency exists.

TABLE II

The upper bound u_d of the frequency of 1s in the Kolakoski sequence induced by the graph G_d.

$u_2 = 2/3$	≈ 0.666667
$u_3 = 2/3$	≈ 0.666667
$u_4 = 5/9$	≈ 0.555556
$u_5 = 8/15$	≈ 0.533333
$u_6 = 36/69$	≈ 0.521739
$\vdots$	
$u_{32} = 3688655/7375520$	≈ 0.500121
$u_{33} = 3845003/7688497$	≈ 0.500098
$u_{34} = 455920839/911696379$	≈ 0.500080

4. Summary

The first algorithm presented to generate the Kolakoski sequence K is highly memory efficient, but runs in linear time and it does not say anything of the long range order of the letter frequency. The ideas to generate K with graphs are much faster and give bounds on the long range letter frequency. The drawback here is that they require an exponential amount of space. Clearly, none of the above described methods settle any part of the Conjecture, they only serve as indications that it should hold. A next step would be to look at properties of the graphs to understand their structure, but this seems to be a very hard and intricate problem.

Acknowledgments

The author wishes to thank V. Terauds, B. Sing, and G. Maloney for our discussions of the subject. This work was supported by the German Research Foundation (DFG) within the CRC 701.

References

[1] R. Oldenburger, *Trans. Am. Math. Soc.* **46**, 453 (1939).

[2] W. Kolakoski, *Am. Math. Monthly* **72**, 674 (1965).

[3] W. Kolakoski, *Am. Math. Monthly* **73**, 681 (1966).

[4] N.J.A. Sloane, *The On-Line Encyclopedia of Integer Sequences, oeis.org*.

[5] C. Kimberling, faculty.evansville.edu .

[6] F.M. Dekking, "Regularity and irregularity of sequences generated by automata", (exposé no. 9) *Sém. Th. Nombres Bordeaux*, 1979–1980, p. 901.

[7] F.M. Dekking, "On the structure of self generating sequences", (exposé no. 31), *Sém. Th. Nombres Bordeaux*, 1980–1981, p. 3101.

[8] F.M. Dekking, *The Mathematics of Long-Range Aperiodic Order, NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.* **489**, 115 (1997).

[9] B. Sing, *More Kolakoski sequences*, arXiv:1009.4061.

[10] J. Nilsson, *J. Integer Seq.* **15**, no. 6 (2012).

[11] M. Rao, www.arthy.org .

[12] V. Chvátal, "Notes on the Kolakoski sequence", *DIMACS Technical Report* 93-84 (1994).

[13] K. Culik, II, J. Karhumäki, A. Lepistö, in: *Lindenmayer Systems*, Eds. G. Rozenberg, A. Salomaa, Springer, Berlin 1992, p. 93.