Rational complexity of binary sequences, FQSRs, and pseudo-ultrametric continued fractions in $\mathbb{R}$

Michael Vielhaber¹ · Mónica del Pilar Canales Chacón² · Sergio Jara Ceballos³

Received: 28 November 2020 / Accepted: 25 September 2021 / Published online: 15 November 2021
© The Author(s) 2021

Abstract

We introduce *rational complexity*, a new complexity measure for binary sequences. The sequence $s \in B^\omega$ is considered as binary expansion of a real fraction $s \equiv \sum_{k \in \mathbb{N}} s_k 2^{-k} \in [0, 1] \subset \mathbb{R}$. We compute its continued fraction expansion (CFE) by the *Binary CFE Algorithm*, a bitwise approximation of s by binary search in the encoding space of partial denominators, obtaining rational approximations r of s with $r \rightarrow s$. We introduce *Feedback in $\mathbb{Q}$ Shift Registers* (FQSRs) as the analogue of Linear Feedback Shift Registers (LFSRs) for the linear complexity L , and Feedback with Carry Shift Registers (FCSR) for the 2-adic complexity A . We show that there is a substantial subset of prefixes with “typical” linear and 2-adic complexities, around $n/2$, but low rational complexity. Thus the three complexities sort out different sequences as non-random.

Keywords Rational complexity · Feedback in $\mathbb{Q}$ Shift Register · FQSR · Binary Continued Fraction Expansion Algorithm · Linear complexity · 2-adic complexity

Mathematics Subject Classification (2010) 94A55 · 68Q19 · 68P30

Notations

$\mathbb{N} = \{1, 2, 3, \dots\}$, $\mathbb{N}_0 = \{0, 1, 2, 3, \dots\}$

$\mathbb{D} := \{a/2^k : a \in \mathbb{Z} \text{ odd}, k \in \mathbb{N} \text{ or } a \in \mathbb{Z}, k = 0\}$, dyadic fractions

$\mathbb{R}_1, \mathbb{Q}_1, \mathbb{D}_1$: For $X \subset \mathbb{R}$, $X_1 := X \cap [0, 1]$

$B = \{0, 1\}$ is the binary alphabet, with complement $\bar{a} = 1 - a$, $a \in B$, e.g. $\overline{10011} = 01100$

$B^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$ and B^ω , are the finite, resp. infinite, words over B

✉ Michael Vielhaber
vielhaber@gmail.com

Mónica del Pilar Canales Chacón
monicadelpilar@gmail.com

Sergio Jara Ceballos
serjara.ing.mat@gmail.com

¹ FB 2, HS Bremerhaven, Bremerhaven, Germany

² @mathupdate, MATEMATICVM, Valdivia, Chile

³ Facultad de Ingeniería, Universidad Austral de Chile, Valdivia, Chile

$B^+ = B^* \setminus \{\varepsilon\}$. For $v = v_1 v_2 \dots v_{|v|} \in B^*$: $(v)_2 = \sum_{k=0}^{|v|-1} v_{|v|-k} 2^k \in \mathbb{N}_0$ in binary, $(\varepsilon)_2 = 0$

1 Introduction and motivation

Stream ciphers employ pseudorandom sequences as a replacement for one-time-pads. Such a cipher is deterministic, but should be indistinguishable from true randomness. A means to assess the quality of such a pseudorandom symbol stream $s \in B^\omega$ is the linear complexity profile (LCP), $(L(s, n), n \in \mathbb{N}_0)$, among other tests. The expected value is $L(s, n) \approx n/2$, and while deviations of $\pm \log_2(n)$ are typical, values with $|L(s, n) - n/2| > 2 \cdot \log_2(n)$ indicate atypical “nonrandom” behaviour, see [15].

Klapper and Goresky introduced 2-adic complexity with profile $(A(s, n), n \in \mathbb{N}_0)$, $A = 1 + \lfloor \log_2(\Phi) \rfloor \in \mathbb{N}$ (see [8, p. 128] for Φ). They demand [6, p. 1]: “*In general, one must consider this ‘2-adic span’ as a measure of security along with ordinary linear span.*”

Linear complexity uses Linear Feedback Shift Registers (LFSRs) and approximates $G(s) = \sum_{k \in \mathbb{N}} s_k x^{-k} \in \mathbb{F}_2[[x^{-1}]]$ by rational functions from $\mathbb{F}_2(x)$, while 2-adic complexity employs Feedback with Carry Shift Registers (FCSR) and approximates $\sum_{k \in \mathbb{N}} s_k 2^{k-1} \in \mathbb{Z}_2$ by rational numbers $p/q \in \mathbb{Q}$ with odd denominator.

The new measure of rational complexity, the corresponding register type Feedback in $\mathbb{Q}$ Shift Register (FQSR), and thus this paper grew out of 3 observations:

- Binary sequences can be interpreted as real numbers from $[0, 1]$ and approximated by rationals from $\mathbb{Q}_1$. No sequence measure seems to treat this point-of-view yet.
- Approximations of $s \in \mathbb{R}_1$ by $P/Q \in \mathbb{Q}_1$ are notoriously difficult numerically due to the Archimedean, non-ultrametric nature of $\mathbb{R}$. Only recently, we developed a pseudo-ultrametric continued fraction expansion (CFE) algorithm which produces about one bit of CFE encoding for each additional bit of the sequence prefix, thus mimicking the Berlekamp-Massey algorithm (BMA) for L in this respect, see [22].
- Comparing linear, 2-adic and rational complexities $L(s, n)$, $A(s, n)$ and $R(s, n)$ of finite prefixes of a certain length n , it turns out that the measures assign low complexities to very different parts of the prefix set:

Motivating Example Consider the three binary prefixes of length $n = 24$ given in Table 1. The columns L, A, R give the linear, 2-adic, and rational complexities, respectively.

As can be seen, in each case two complexities lie around $n/2 = 12$, which is near the expected average behaviour (see [15] for L , [16] for A : we only know $(n-1)/2 \leq E(A) < 0.772 \cdot n$, and Section 5 for R), and thus the sequence is unremarkable from that point of view. Each sequence, however, has a markedly *low* value in one of the complexities, (1) for L , (2) for A , and (3) for R . Hence, from a cryptographic point of view, all three sequences should be discarded as non-pseudorandom. We need though all three tests, L , A , and R to encounter all three of them as atypical.

Table 1 Prefixes with distinct complexities L, A, R

	Sequence prefix	Complexities		
		L	A	R
(1)	1011.0010.0011.1101.0110.0100	4	12	14
(2)	0110.1001.1101.1110.0101.1000	12	5	14
(3)	0100.1111.0111.0010.1100.0010	12	14	5

Like Klapper stated for the 2-adic complexity, we therefore should also consider the rational complexity R , defined in Section 5, as a means to assess randomness.

The general idea is to represent a given binary sequence as a real number

$$s = \sum_{k \in \mathbb{N}} s_k 2^{-k} \in \mathbb{R}_1 \subset \mathbb{R}$$

and to obtain its CFE, and thereby good rational approximations. By reuse of notation, we will write s both for the binary sequence and the resulting real number.

Going in the direction from s to its CFE is a very tedious process over $\mathbb{R}$. It is not suited to bitwise, incremental computation with growing prefix length, since it needs the “full” precision (whatever that will be) right from the start. Instead, we define a sequence of CFEs (of rational numbers) that converge to the CFE of s , by doing a binary search in the space of all CFEs. Due to the results from [22], this is possible and efficient.

We obtain an encoding of the CFEs which is both monotonic

$$s < s' \iff \text{encoding}(\text{CFE}(s)) < \text{encoding}(\text{CFE}(s'))$$

(in $\mathbb{R}$ and in lexicographical order, respectively) and at the same time pseudo-ultrametric: If two sequences (or binary representations of numbers) s and s' differ in bit k , but not before, their encodings differ—in the limit for large k , for numbers s, s' satisfying the so-called Gauß-Kuz'min measure (Section 3)—for the first time around bit $\lfloor 1.024 \cdot k \rfloor$, not much earlier and not much later: The information content provided by $s_1, \dots, s_k$ is roughly preserved and translated into $\approx 1.024 \cdot k$ bits of information about the CFE of s .

The actual trial values $r \in \mathbb{Q}_1$ with $r \rightarrow s$ are taken from the so-called V_{10} tree [22]. Let v be the address of r within this infinite binary tree ($v = \varepsilon$ at the root, each move down appends a bit to v , 0 to the left, 1 to the right). Then the CFE encoding of r will be $v10^\omega \in B^\omega$.

Therefore apparently, since we move further and further down the V_{10} tree, any address is a prefix of the CFE encoding of s : the prefix grows when moving down, previously defined bits are not altered afterwards.

This paper is divided in 3 parts.

First, we treat continued fractions (Section 2), encodings (Section 3), the Binary CFE algorithm (Section 4), define Rational Complexity (Section 5), and give an example.

The second part treats properties of rational complexity such as connections with 2-adic complexity for periodic sequences as well as impulse response sequences (Section 7). A comparison of the three complexity measures R , L , and A on 30-bit prefixes (a generalisation of the “Motivating Example”) is given in Section 8. We show that encodings and prefix lengths do not differ too much (the “pseudo-ultrametric” property) in Section 9. In Section 10, we introduce the Q Tree, providing an ultrametric ordering of $\mathbb{Q}_1$.

The final part covers implementation aspects: In Section 11, we show, how to efficiently update the convergents P/Q , when moving down the V_{10} tree, by a Finite State Machine with Counter. Section 12 covers Feedback in $\mathbb{Q}$ Shift Registers (FQSR) which allow to efficiently compute the infinite rational sequence r —to be compared to the input s —from the two numbers P and Q . We finish with Open Problems.

Table 2 Schema for PDs b_i and convergents P_i/Q_i , for $s = \pi - 3$

i	-2	-1	0	1	2	3	4	...
b_i			0	7	15	1	292	...
P_i	0	1	0	1	15	16	4687	...
Q_i	1	0	1	7	106	113	33102	...

2 Continued fraction expansions in $\mathbb{R}$ and $\mathbb{F}_2[[x^{-1}]]$

Case $\mathbb{R}$ The usual way to compute the CFE of a number $s := \sum_{k \in \mathbb{N}} s_k 2^{-k} \in \mathbb{R}_1$ is

$$s^{(0)} := s, b_0 := 0, \forall i \in \mathbb{N}: s^{(i)} := \frac{1}{\{s^{(i-1)}\}} = \frac{1}{s^{(i-1)} - b_{i-1}}, b_i := \lfloor s^{(i)} \rfloor,$$

giving $s = 0 + \frac{1}{|b_1|} + \frac{1}{|b_2|} + \dots$, where the resulting b_i are the partial denominators (PDs).

We write $s = [b_1, b_2, \dots]$ for the CFE.

The convergents P_i/Q_i to s are obtained via Perron's schema, Table 2 (see [14], we use $s = \pi - 3$ as example). The initial values are $P_{-2} = Q_{-1} = 0$, $Q_{-2} = P_{-1} = 1$, and then $P_i := b_i \cdot P_{i-1} + P_{i-2}$, $Q_i := b_i \cdot Q_{i-1} + Q_{i-2}$ with $|s - P_i/Q_i| < Q_i^{-2}$ for $s \notin \mathbb{Q}$, [14, Satz 2.10].

Case $\mathbb{F}_2[[x^{-1}]]$ In the case of formal power series from $\mathbb{F}_2[[x^{-1}]]$, one computes the linear complexity via the Berlekamp-Massey Algorithm (BMA) [1, 11]. The BMA takes the sequence s of coefficients and computes the CFE

$$\mathbb{F}_2[[x^{-1}]] \ni G(s) = \sum_{k=1}^{\infty} s_{kx}^{-k} = \frac{1}{|p_1(x)|} + \frac{1}{|p_2(x)|} + \frac{1}{|p_3(x)|} + \dots,$$

with polynomials $p_i(x) \in \mathbb{F}_2[x]$ as partial denominators. With the adaptations by Dornstetter [3] and Vielhaber [19], the BMA computes an isometry $\mathbf{K}$ on $\mathbb{F}_2^\omega$ such that the modified discrepancy sequence $d := \mathbf{K}(s)$ (see [19])

$$\mathbf{K}(s) = d = (d_k) = C_{\mathbb{F}_2[x]}(p_1) | C_{\mathbb{F}_2[x]}(p_2) | C_{\mathbb{F}_2[x]}(p_3) | \dots \in \mathbb{F}_2^\omega$$

is at the same time an encoding of the PDs, by

$$C_{\mathbb{F}_2[x]}: \mathbb{F}_2[x] \setminus \mathbb{F}_2 \rightarrow \mathbb{F}_2^*, \quad p(x) = \sum_{k=0}^g a_k x^k \mapsto C_{\mathbb{F}_2[x]}(p(x)) = 0^{g-1} 1 a_{g-1} \dots a_1 a_0.$$

The degrees $g_i = \deg(p_i)$ of the PDs determine the linear complexity profile of s (see [19]). For fast implementations see [2, 13].

Our approach is inspired by this ultrametric, bitwise model and will avoid the costly inversions $s^{(i)} := 1/\{s^{(i-1)}\}$ (which would have to be executed each time with the maximum prefix length corresponding to the eventually desired precision).

3 Binary CFE I: Gauß-Kuz'min measure, codes, monotonicity

3.1 Gauß-Kuz'min measure

To define an encoding for PDs, we start with the relevant facts obtained by Gauß, Kuz'min, Khinchin, and Lévy about continued fractions for $r \in \mathbb{R} \setminus \mathbb{Q}$ and their PDs b_i . The probability for a PD value of b is $\mu_{GK}(b) := \lim_{n \rightarrow \infty} |\{i \mid 1 \leq i \leq n, b_i = b\}|/n$.

Theorem 1 Gauß-Kuz'min-Khinchin-Lévy [5, 9, 10]

For almost all values $r \in \mathbb{R}$ (exceptions are rational, quadratic-algebraic and some more numbers, including $e^{2/k}$, $k \in \mathbb{N}$), we have:

- (i) The probability for a partial denominator $b \in \mathbb{N}$, its Gauß-Kuz'min measure, is

$$\mu_{GK}(b) = -\log_2 \left(1 - \frac{1}{(b+1)^2} \right) = \log_2 \left(1 + \frac{1}{b(b+2)} \right).$$

- (ii) The geometric average of the partial denominators is Khinchin's constant

$$K := \lim_{n \rightarrow \infty} \sqrt[n]{b_1 \cdot b_2 \cdots b_n} \doteq 2.68545.$$

- (iii) The average gain in precision, per partial denominator in bits, is

$$\frac{\pi^2}{6 \ln(2)^2} \doteq 3.42371, \text{ with } 2^{3.42371/2} \doteq 3.27582 \text{ being Lévy's constant.}$$

3.2 Encodings C_I , C_{II} , and C

The encoding $C: \mathbb{R}_1 \rightarrow B^\omega \setminus B^{*1}{}^\omega$ for CFEs, i.e. for the sequence of PDs from $\mathbb{N}$, shall satisfy the following 3 desiderata:

(GK) C is reasonably close to the Gauß-Kuz'min measure.

We want to be almost isometric that is the encoding length should advance in parallel with the sequence prefix, bit-by-bit (preservation of information content). The measure $\mu_{GK}(b)$ for the expected fraction of the PD value $b \in \mathbb{N}$ thus requires that an encoding for b has length around $-\log_2(\mu_{GK}(b))$.

(MON) C is monotonically increasing: $r < r' \iff C(r) < C(r')$ in lexicographical order.

This allows to avoid the costly CFE computation for the real s altogether. Instead, we do a binary search in the encoding space $C(\mathbb{R}_1) \subset B^\omega$ and compare the resulting (rational!) number r with s , where (MON) tells us, which of the two subintervals of the binary search remains valid, and (GK) ensures efficiency for this process.

(INC) C is incremental: Given the last two convergents P/Q and P'/Q' , from $r = P/Q = C^{-1}(va10^\omega)$ we obtain

$$C^{-1}(va10^\omega) = \frac{P \pm (P' \cdot 2^{\delta_a})}{Q \pm (Q' \cdot 2^{\delta_a})} \text{ or } C^{-1}(va10^\omega) = \frac{2P + P'}{2Q + Q'}$$

for both values $a \in B$ with some $\delta_a \in \mathbb{N}_0$. This yields an efficient implementation, where the multiplication $P_i := b_i \cdot P_{i-1} + P_{i-2}$ is replaced by iterated shift-and-add. This requirement (INC) is only relevant for fast implementations, see Section 11.

Table 3 Codes C_I and C_{II} for partial denominators

b	C_I	C_{II}	$l_{I,II}$	l_{GK}	μ_{GK}
1	1	0	1	1.269	0.4150
2	011	100	3	2.557	0.1699
3	010	101	3	3.425	0.0931
4	00111	11000	5	4.086	0.0588
5	00110	11001	5	4.621	0.0406
6	00101	11010	5	5.071	0.0297
7	00100	11011	5	5.460	0.0227
8	0001111	1110000	7	5.802	0.0179
9	0001110	1110001	7	6.108	0.0144
10	0001101	1110010	7	6.384	0.0119
11	0001100	1110011	7	6.636	0.0100
12	0001011	1110100	7	6.868	0.0085
13	0001010	1110101	7	7.082	0.0073
14	0001001	1110110	7	7.282	0.0064
15	0001000	1110111	7	7.468	0.0056
16	000011111	111100000	9	7.644	0.0050
⋮					
31	000010000	111101111	9	9.471	0.0014
32	00000111111	11111000000	11	9.559	0.0013
⋮					
63	00000100000	11111011111	11	11.471	0.00035
64	0000001111111	1111110000000	13	11.516	0.00034
⋮					
$\aleph_0$	0^ω	1^ω	–	–	–

Definition 2 Encodings C_I , C_{II} , and C (Table 3)

We encode a PD $b \in \mathbb{N}$, $b = 2^n + \sum_{k=0}^{n-1} a_k 2^k$ with $n := \lfloor \log_2(b) \rfloor$ as follows:

- (i) For PDs b_i with odd index i , let

$$C_I: \mathbb{N} \rightarrow B^*, b \mapsto C_I(b) = 0^n 1 \bar{a}_{n-1} \dots \bar{a}_1 \bar{a}_0.$$

In particular, $1 \mapsto 1$; $2 \mapsto 011$; $3 \mapsto 010$. We also use the “pseudo-PD” $\aleph_0$ (Aleph-Zero), the first, countable infinity, with $1/\aleph_0 := 0$, and set $C_I: \aleph_0 \mapsto 0^\omega$.

- (ii) For PDs b_i with even index i , we complement the codewords to obtain code C_{II} :

$$C_{II}: \mathbb{N} \rightarrow B^*, b \mapsto C_{II}(b) = \overline{C_I(b)} = 1^n 0 a_{n-1} \dots a_1 a_0.$$

In particular, $1 \mapsto 0$; $2 \mapsto 100$; $3 \mapsto 101$; $4 \mapsto 11000$, and $C_{II}(\aleph_0) := 1^\omega$.

Observe that both $C_{\mathbb{F}_2[\aleph]}$ and C_I, C_{II} are composed of a first half, encoding the size (degree / $\lfloor \log_2(b) \rfloor$), and a second half, describing the details from “coarse”

$(a_{d-1}a^{d-1} / a_{n-1}2^{n-1})$ to “fine” (a_0x^0 / a_02^0) . C_I and C_{II} are prefixfree and complete prefixcodes.

(iii) We concatenate encodings (rational finite CFE and irrational infinite CFE) as

$$C: \mathbb{Q}_1 \rightarrow B^*0^\omega, r = [b_1, b_2, \dots, b_{2l}] \mapsto C(r) := C_I(b_1)|C_{II}(b_2)|\dots|C_{II}(b_{2l})|C_I(\aleph_0).$$

and

$$C: \mathbb{R} \setminus \mathbb{Q}_1 \rightarrow B^\omega \setminus (B^*0^\omega \cup B^*1^\omega), r = [b_1, b_2, b_3, \dots] \mapsto C(r) := C_I(b_1)|C_{II}(b_2)|C_I(b_3)|\dots$$

About C for $\mathbb{Q}_1$: A finite CFE can be represented in two ways, as $[\dots, b_n, 1] = [\dots, b_n + 1]$, leading to the same value. We can resolve this ambiguity in 4 ways:

- (i) Let the last PD always be greater than 1, or
- (ii) always equal to 1, or
- (iii) always have an even, or (iv) always an odd number of PDs.

We use convention (iii), thus C always terminates in some $C_{II}(b_{2l})$, followed by the “pseudo-PD” $\aleph_0$ with $C_I(\aleph_0) = 0^\omega$. We therefore have $B^*0^\omega = C(\mathbb{Q}_1)$ as encodings of finite CFEs.

(GK) Columns $l_{I,II}$, with $l_{I,II}(b) = 1 + 2\lfloor \log_2(b) \rfloor$, and l_{GK} , with $l_{GK}(b) = -\log_2(\mu_{GK}(b))$, of Table 3 give the coding length for our code and an “ideal code” according to the Gauß-Kuz’min measure (which would lead to non-integral word lengths), respectively. As can be seen, the two columns have reasonably similar values for each b . The codes are from [22], but C_{II} already appears in Vallée [17] and she mentions Elias (reference [12] in [17]) as inventor. The complemented code C_I and the interleaving as C are though apparently new.

Comparing with L , the analogous measure for a polynomial PD $p(x) = x^m + \sum_{k=0}^{m-1} a_k x^k \in \mathbb{F}_2[x]$ is $\mu(p) = 2^{-2m}$, and the encoding $\mathbf{K}$ (see [19]) is an isometry, with $|\mathbb{C}_{\mathbb{F}_2[x]}(p)| = 2m = -\log_2(\mu(p))$. In a sense, this paper tries to achieve the same in the setting of $\mathbb{R}$.

3.3 Monotonicity

We next show the monotonicity of $C: \mathbb{R} \rightarrow B^\omega$, where $<$ is the usual order relation in $\mathbb{R}$, and refers to lexicographical order in B^ω . From continued fraction theory, we know the impact on r of changing a PD b_i :

Proposition 3 *Let two real numbers $r' = [b_1, \dots, b_{i-1}, b'_i, \dots]$, $r'' = [b_1, \dots, b_{i-1}, b''_i, \dots]$ with $b'_i < b''_i$ be given. Then $r' < r''$ for even index i , and $r' > r''$ for odd i , respectively.*

Proof See Satz 2.8 in Perron [14]. □

Theorem 4 (MON) *Monotonicity of $C(r)$*

Let $r', r'' \in [0, 1) \subset \mathbb{R}$ with $r' < r''$, and $C(r')$, $C(r'')$ their encodings.

Then $C(r') < C(r'')$ in lexicographical order.

Proof Follows with Proposition 3. As mentioned (see [14, Satz 2.8]), for odd index $2i + 1$ (b_j , $j < 2i + 1$, fixed), $b'_{2i+1} > b''_{2i+1}$ implies $r' = [b_1, \dots, b_{2i}, b'_{2i+1}] < r'' = [b_1, \dots, b_{2i}, b''_{2i+1}]$. Also, we then have $C_I(b'_{2i+1}) < C_I(b''_{2i+1})$ and thus $C(r') < C(r'')$ lexicographically.

For even index $2i + 2$ (b_j , $j < 2i + 2$, fixed), we have $b'_{2i+2} < b''_{2i+2} \Leftrightarrow r' < r'' \Leftrightarrow C_{II}(b'_{2i+2}) < C_{II}(b''_{2i+2}) \Leftrightarrow C(r') < C(r'')$. □

Table 4 Binary CFE and approximations

	$v.10^\omega$	PDs	P/Q	$r = .x(y) := xy^\omega$
Binary CFE	$\varepsilon.1 0 $	[1, 1]	1/2	.1(0)
$v := \varepsilon$ // <i>code</i> ($[b_i]$)	0.10 0	[3, 1]	1/4	.01(0)
$s := \varepsilon$ // <i>input sequence</i>	1. 100	[1, 2]	2/3	.(10)
$r := 10^\omega$ // <i>approximation</i>	00.100 0	[7, 1]	1/8	.001(0)
// $r = C^{-1}(v10^\omega)$	01.1 0	[2, 1]	1/3	.(01)
$d := \varepsilon$ // <i>discrepancies</i>	1 0. 1 0	[1, 1, 1, 1]	3/5	.(1001)
// $d = \mathbf{D}(s)$	1 1.1000	[1, 4]	4/5	.(1100)
$k := 0$ // <i>index in s, d, r</i>	000.1000 0	[15, 1]	1/16	.0001(0)
FOREVER	001.10 0	[5, 1]	1/6	.0(01)
$k := k + 1$	010. 100	[3, 2]	2/7	.(010)
read_next_bit s_k	011. 100	[2, 2]	2/5	.(0110)
IF $s_k = r_k$	1 0 0.10 0	[1, 1, 3, 1]	5/9	.(100011)
// Case ($\vee$)	1 0 1. 100	[1, 1, 1, 2]	5/8	.101(0)
$d := d 0$	1 10.1	[1, 3]	3/4	.11(0)
ELSE	1 11.10000	[1, 8]	8/9	.(111000)
$d := d 1$	0000.10000 0	[31, 1]	1/32	.00001(0)
WHILE $(s_1..s_k) \neq (r_1..r_k)$	0001.100 0	[11, 1]	1/12	.00(01)
IF $(s_1..s_k) < (r_1..r_k)$	0010.1 0	[6, 1]	1/7	.(001)
// Case $\ominus$	0011.1 0	[4, 1]	1/5	.(0011)
$v := v 0$	010 0. 1 0	[3, 1, 1, 1]	3/11	.(0100010111)
ELSE $(s_1..s_k) > (r_1..r_k)$	010 1.1000	[3, 4]	4/13	.(010011101100)
// Case $\oplus$	011 0. 1 0	[2, 1, 1, 1]	3/8	.011(0)
$v := v 1$	011 1.1000	[2, 4]	4/9	.(011100)
ENDIF	1 0 00.100 0	[1, 1, 7, 1]	9/17	.(10000111)
$[b_i] := \text{decode}(v10^\omega)$	1 0 01.1 0	[1, 1, 2, 1]	4/7	.(100)
$(P, Q) := \text{perron}([b_i])$	1 0 1 0. 1 0	[1, 1, 1, 1, 1, 1]	8/13	.(100111011000)
$r := P/Q$	1 0 1 1.1000	[1, 1, 1, 4]	9/14	.1(010)
ENDWHILE	1 100. 1 0	[1, 2, 1, 1]	5/7	.(101)
ENDIF	1 101. 1 0	[1, 3, 1, 1]	7/9	.(110001)
ENDFOR	1 110.10	[1, 6]	6/7	.(110)
	1 111.0000	[1, 16]	16/17	.(11110000)

4 Binary CFE II: Algorithm

The Binary Continued Fraction Expansion Algorithm (Table 4, left) builds upon the monotonicity of $C: \mathbb{R}_1 \rightarrow B^\omega$ from sequences to encodings. We use the computationally easier inverse to obtain a trial value $r \in \mathbb{Q}_1$ from an encoding with finite support in B^*10^ω , $r := C^{-1}(v10^\omega)$, $v \in B^*$. We then compare s to r with $s < r \Leftrightarrow C(s) < C(r)$, Theorem 4.

The function C^{-1} restricted to B^*10^ω is represented in the V_{10} Tree defined in [22], an infinite binary tree with labels, the binary analogue to the Stern-Brocot Tree. Table 4 (right) (see also Table 9) gives the first 5 levels of the V_{10} Tree, and therefore the first 5 trial values (with $|v| \leq 4$) for any sequence s , with $C^{-1}(v10^\omega) = [\text{PDs}] = P/Q = r \in \mathbb{Q}_1$.

The $|$ separates the encodings C_I , C_{II} and the “.” stands after the current prefix v .

The V_{10} Tree arranges the $r \in \mathbb{Q}_1$ according to their CFE support length. The r with CFE $v10^\omega$ appears as label at address v on level $|v| + 1$, level 1 being the root node with address ε . The node at v has child nodes with addresses $v0$ (left) and $v1$ (right child).

Moving down the V_{10} Tree with growing v , we successively shrink the set vB^ω of permissible encodings, whose center is the trial encoding $v10^\omega$. Analogously to the BMA for L , we compute a discrepancy sequence $\mathbf{D}(s) = (d_k)$, where $d_k = 0$, if the approximation of

s by r up to bit $k - 1$ also satisfies $s_k = r_k$; $d_k = 1$ otherwise, for $s_k \neq r_k$. By definition, $\mathbf{D}$ is again an isometry on B^ω .

The bits within the sequences, input s , trial value r , discrepancies d , are indexed by k and the index is advanced in the FOREVER loop as long as $s_k = r_k$ with $d_k = \mathbf{D}(s)_k := 0$.

Whenever a discrepancy $r_k \neq s_k$, $d_k = \mathbf{D}(s)_k := 1$ occurs, we enter the WHILE loop to adjust the encoding, the CFE, and thereby the trial value r :

Either we have $(s_1 \dots s_k) < (r_1 \dots r_k)$ lexicographically, and r is too large, has to decrease, Case $\ominus$: We set $v := v0$, leading to $v0B^\omega$ as new cylinder set of valid encodings, with the previous trial encoding $v10^\omega$ as upper bound. Thus $v010^\omega$ is the new trial encoding—the new center value.

Or, to the contrary, we had $(s_1 \dots s_k) > (r_1 \dots r_k)$, Case $\oplus$: Then we adjust to $v := v1$, with $v110^\omega$ the center of $v1B^\omega$. This interval halving or binary search within $B^\omega = C(\mathbb{R}_1)$ allows to avoid the costly inversions $s^{(i)} := 1/\{s^{(i-1)}\}$ (see Section 2, CFE in $\mathbb{R}$), which in principle would have to be carried out with the full precision of $|s|$.

Given the new v (one new bit appended to the PD encoding), we generate the new convergent $r = P/Q$ incrementally. We have to take three actions:

- I $v \mapsto v10^\omega \xrightarrow{C^{-1}} [b_1, b_2, \dots, b_{2l}, \aleph_0]$, obtain the new sequence of PDs.
- II $[b_1, b_2, \dots, b_{2l}] \mapsto P/Q$, evaluate the encoded PDs by Perron's schema.

For now, cut $v10^\omega$ into the pieces according to C_I, C_{II} in I, and then use Table 2 for II. Section 11 will show, how the new convergents P/Q are obtained almost effortlessly from the two previous ones, combining I and II, by property (INC).

- III $P/Q \mapsto r \in B^\omega$, expand in binary. Again, for now think “paper-and-pencil long division in binary / in base 2”. Section 12 will introduce FQSRs to do this efficiently.

If now $s_i = r_i$, $1 \leq i \leq k$, we leave the WHILE loop and return into the FOREVER loop, otherwise we refine the PD encoding by again extending v as long as necessary.

For sequences satisfying Theorem 1 (Gauß-Kuz'min), we enter the WHILE loop on average every second cycle of the FOREVER loop (for $s_k \neq r_k$ i.e. $d_k = 1$), and we show in Section 9 that $\lim_{k \rightarrow \infty} |v|/k \doteq 1.024$ in this case. Therefore, we stay within the WHILE loop for an average of 2.048 cycles for such sequences.

5 Rational complexity of a binary sequence

Before defining rational complexities, a comment is in order: The reals $\mathbb{R}_1$, and B^ω as their dyadic representations, are not topologically equivalent. Rather B^ω is isomorphic to the Cantor set (see Vepstas [18]). As a consequence, a dyadic fraction $r \in \mathbb{D}_1 \subset \mathbb{Q}_1$ corresponds to the two sequences $w10^\omega$ and $w01^\omega$ for some $w \in B^*$. We set the complexity for $w01^\omega$ to at most $|w| + 2$, including the first 1 of the “stuck-at” part in the length.

For $2^{-k} = 0^{k-1}10^\omega = 0^k1^\omega$, this leads to $R = k$ and $k + 1$, respectively. Apparently, for the latter, a k -bit register would be initialized with 0^k and could not generate 0^k1^ω .

We now define the rational complexity of a sequence in analogy to L and A .

Definition 5 Rational Complexity $R(s, n)$

- (i) Let $R(s, 0) := 0, \forall s \in B^* \cup B^\omega$. For all zero sequences, let $R(0^\omega, n) := 0, n \in \mathbb{N}$ and $R(0^k, n) := 0, 1 \leq n \leq k$.
- (ii) For sequences $s \in B^* \setminus (\{0\}^* \cup B^*11)$ i.e. neither allzero, nor ending in 11, and for $s \in B^\omega \setminus (\{0^\omega\} \cup B^*1^\omega)$, the rational complexity of s is

$$R(s, n) := \lceil \log_2(Q_i) \rceil,$$

where $P_i/Q_i = r$ is the first convergent of the Binary CFE matching $r_k = s_k, 1 \leq k \leq n$. In view of Section 12, $R(s, n)$ is the length of an FQSR producing the n -bit prefix of s .

- (iii) For sequences $s = v01^k, k \geq 2$ or $s = v01^\omega$ with some $v \in B^*$, let $R(s, n) = \min(|v| + 2, \lceil \log_2(Q_i) \rceil), |v| + 2 \leq n \leq |s|$, with Q_i as in (ii). Also, $R(1^\omega, n) := 1, \forall n \in \mathbb{N}$.
- (iv) Let the rational complexity profile for a sequence $s \in B^*$ be $(R(s, n), 0 \leq n \leq |s|)$. For an $s \in B^\omega$, the profile is $(R(s, n), n \in \mathbb{N}_0)$.
- (v) For ultimately periodic sequences $s = x(y)^\omega$ with $x \in B^*$ and $y \in B^+$, let $R(s) := \lim_{n \rightarrow \infty} R(s, n)$. Then $R(s)$ is the length of the shortest FQSR producing s .
- (vi) The number of sequences of length n with rational complexity $c \in \mathbb{N}_0$ is denoted by $N_R(n, c)$, and we set $N_R(c) := \lim_{n \rightarrow \infty} N_R(n, c)$.

Let $E(R(n)) := 2^{-n} \cdot \sum_{s \in B^n} R(s, n) = 2^{-n} \cdot \sum_{k=0}^n k \cdot N_R(n, k)$ be the average rational complexity for words of length n , and $\text{Var}(R(n))$ the corresponding variance. The distribution of rational complexities for words of length $n \leq 12$, as well as $E = E(R(n))$ and $\text{Var} = \text{Var}(R(n))$ for $n \leq 24$ are given in Table 5.

Table 5 Rational complexity: $N_R(n, c)$, $E(R(n))$, $\text{Var}(R(n))$

n	E	Var	c :	0	1	2	3	4	5	6	7	8	9	10	11	12
1	0.500	0.2500		1	1											
2	1.000	0.5000		1	2	1										
3	1.750	0.9375		1	2	3	2									
4	2.438	1.1211		1	2	4	7	2								
5	2.969	1.1553		1	2	5	15	7	2							
6	3.609	1.2068		1	2	5	17	29	8	2						
7	4.227	1.2846		1	2	5	18	49	41	10	2					
8	4.805	1.2978		1	2	5	18	59	114	43	12	2				
9	5.303	1.2619		1	2	5	18	62	213	157	40	12	2			
10	5.825	1.2302		1	2	5	18	62	248	479	151	44	12	2		
11	6.374	1.2693		1	2	5	18	62	252	793	676	177	48	12	2	
12	6.892	1.2556		1	2	5	18	62	252	942	1835	731	186	48	12	2
n	E	Var	n	E	Var	n	E	Var	n	E	Var					
13	7.405	1.2602	16	8.924	1.2475	19	10.434	1.2572	22	11.935	1.2479					
14	7.913	1.2537	17	9.429	1.2571	20	10.933	1.2479	23	12.437	1.2571					
15	8.420	1.2596	18	9.929	1.2473	21	11.437	1.2574	24	12.935	1.2476					

Theorem 6 Rational Complexity: Counts

The number $N_R(c)$ is $N_R(0) = 1$ (for $s = 0^\omega$), $N_R(1) = 2$ (for $s = 10^\omega, 1^\omega$), and for $c \geq 2$

$$N_R(c) = 2^{c-2} + \sum_{Q=2^{c-1}+1}^{2^c} \varphi(Q),$$

where φ is Euler's totient function.

Proof The sum is the number of fractions P/Q with $2^{c-1} < Q \leq 2^c$, $1 \leq P < Q$ and P, Q coprime. The additional 2^{c-2} sequences correspond to the special cases $v01^\omega$ with $v \in B^{c-2}$, the set of words over B of length $c-2$, for dyadic fractions (Def. 5 (iii)). $\square$

Conjecture 7 Rational Complexity: Average and Variance

The following limits exist and yield four small constants $v_E^0, v_E^1, v_V^0, v_V^1$:

$$v_E^a := \lim_{\substack{n \rightarrow \infty \\ n \equiv a \pmod{2}}} E(R(n)) - (n/2 + 1) \text{ and } v_V^a := \lim_{\substack{n \rightarrow \infty \\ n \equiv a \pmod{2}}} \text{Var}(R(n)) - 5/4, \quad a \in \{0, 1\}.$$

The approximation of each PD may be alternating from above and below. Thus the rational complexity is not necessarily monotonic (the sequence of encodings $v10^\omega$ though has monotonically increasing support length $|v| + 1$). However, the error in monotonicity is at most one:

Proposition 8 For all sequences $s \in B^\omega$ and all $k_1 < n < k_2 \in \mathbb{N}$, we have

$$R(s, k_1) - 1 \leq R(s, n) \leq R(s, k_2) + 1.$$

Proof A reduction of the absolute value of Q , and thereby possibly of $R(s, n)$, may only occur when the current PD b is decreased.

The PDs first increase through the values $2^k, k = 1, 2, 3, \dots$ with either $C_I(2^k - 1) = 0^{k-1}10^{k-1}$, $C_{II}(1) = 0$ or $C_{II}(2^k) = 1^{k-1}0^k$ as last encodings before $C_I(8_0) = 0^\omega$, see Table 3 (also Section 11, Table 11). They then settle within the interval $[2^n, 2^{n+1}]$ for some n — only this may decrease R .

The claim follows with $Q + Q' \leq Q + 2^n Q' \leq Q + bQ' \leq Q + 2^{n+1} Q' < 2(Q + 2^n Q')$. $\square$

Remark 9 Being monotonic with growing prefix length is of course a desirable property of a complexity measure. We therefore define a monotonic version of R in the next definition. Note, however that this approach is no longer the Binary CFE with incremental bitwise consideration of both the sequence and the encoding.

Definition 10 Monotonic Rational Complexity $\bar{R}$

Let $\bar{R}(s, k) := R(s, k)$ for all sequences s and positions $k \leq |s|$, with two exceptions (following the algorithm from Table 4, left):

- (i) If $R(s, k)$ is the result of the WHILE-loop with Case $\ominus$ in an encoding C_I or a case $(C_{II}, \oplus)$, then $R(s, k)$ was computed from an upper bound for b . In this case, advance the encoding (append to v) until a situation $(C_I, \oplus)$ or $(C_{II}, \ominus)$, a lower bound for b , occurs within the current encoding, or until reaching the encoding of the next PD, irrespective of $\oplus, \ominus$. If hereby the rational complexity decreases to $R(s, k) - 1$, take this value for $\bar{R}(s, k) := R(s, k) - 1$, otherwise, keep $\bar{R}(s, k) = R(s, k)$.

Table 6 Approximating π

Read	S	$v.10^\omega$	$[(b_i)]$	p/q	R	r	k	$\ominus/\oplus$
	A	.1 0	[1, 1]	1/2	1	8000	1	$\ominus$
s_1	B ₁	0.10 0	[3, 1]	1/4	2	4000	2	$\ominus$
s_2	B ₂	00.100 0	[7, 1]	1/8	3	2000	6	$\oplus$
$s_3 \dots s_6$	C ₂	001.10 0	[5, 1]	1/6	(3)	2AAA	5	$\ominus$
	D	0010.1 0	[6, 1]	1/7	3	249249	9	$\ominus$
$s_7 \dots s_9$	$\overline{A}$	00100. 100	[7, 2]	2/15	(4)	2222	6	$\oplus$
	$\overline{B_1}$	00100 1.1000	[7, 4]	4/29	(5)	234	6	$\oplus$
	$\overline{B_2}$	00100 11.10000	[7, 8]	8/57	(6)	23E	6	$\oplus$
	$\overline{B_3}$	00100 111.100000	[7, 16]	16/113	7	243F6F	22	$\ominus$
$s_{10} \dots s_{22}$	$\overline{C_3}$	00100 1110.100	[7, 12]	12/85	(7)	242424	12	$\oplus$
	$\overline{C_2}$	00100 11101.10	[7, 14]	14/99	(7)	2433B	13	$\oplus$
	$\overline{D}$	00100 111011.1	[7, 15]	15/106	(7)	2439F	14	$\oplus$
	A	00100 1110111. 1 0	[7, 15, 1, 1]	31/219	(8)	243CC	15	$\oplus$
	$\overline{A}, \overline{B_{1..6}}$	$\vdots$	[7, 15, 1, 2^{j+1}]			$\vdots$		$\oplus$
	$\overline{B_7}$	00.. ..11 1 1 ⁷ .10 ⁹	[7, 15, 1, 256]		15	243F69E52	23	$\oplus$
s_{23}	$\overline{B_8}$	00.. ..11 1 1 ⁸ .10 ¹⁰	[7, 15, 1, 512]		(16)	243F6C728	22	$\ominus$
	$\overline{C_8}$	00.. ..11 1 1 ⁸ 0.10000000	[7, 15, 1, 384]		16	243F6B987	24	$\ominus$
s_{24}	$\overline{C_7}$	00.. ..11 1 1 ⁸ 00.1000000	[7, 15, 1, 320]		16	243F6AEA3	26	$\ominus$
s_{25}, s_{26}	$\overline{C_6}$	00.. ..11 1 1 ⁸ 000.100000	[7, 15, 1, 288]		(15)	243F6A762	25	$\oplus$
	$\overline{C_5}$	00.. ..11 1 1 ⁸ 0001.10000	[7, 15, 1, 304]		16	243F6AB33	27	$\ominus$
s_{27}	$\overline{C_4}$	00.. ..11 1 1 ⁸ 00010.1000	[7, 15, 1, 296]		16	243F6A958	28	$\ominus$
s_{28}	$\overline{C_3}$	00.. ..11 1 1 ⁸ 000100.100	[7, 15, 1, 292]		16	243F6A860	29	$\oplus$
s_{29}	$\overline{C_2}$	00.. ..11 1 1 ⁸ 0001001.10	[7, 15, 1, 294]		16	243F6A8DD	30	$\ominus$
s_{30}	$\overline{D}$	00.. ..11 1 1 ⁸ 00010010.1	[7, 15, 1, 293]		16	243F6A89F2	32	$\ominus$
s_{31}, s_{32}	A	00.. .. 1 ..100 .1 0	[7, 15, 1, 292, 1, 1]		(17)	243F6A87FF	29	$\oplus$
	$\overline{A}$	00.. .. 1 ..100 1 .100	[7, 15, 1, 292, 1, 2]		17	243F6A88A5	35	$\ominus$
$s_{33} \dots s_{35}$	A	00.. .. 1 ..100 1 0 .1 0	[7, 15, 1, 292, 1, 1, 1, 1]		18	243F6A8863	36	$\ominus$

- (ii) If, in a Case ($\checkmark$), we previously had $\overline{R}(s, k-1) = R(s, k-1) - 1$, keep that value, i.e. $\overline{R}(s, k) := \overline{R}(s, k-1) = R(s, k-1) - 1 = R(s, k) - 1$.

6 Example π

Example 11 The well-known CFE of $\pi \doteq 0x3.243F6A8885A308D$ is [3; 7, 15, 1, 292, 1, ...]. The Binary CFE Algorithm proceeds on $\pi - 3$ as in Table 6. We omit the final code 0^ω and PD $\aleph_0$. $\{S \in A, \dots, \overline{D}\}$ is the state of the FSM+C (Section 11). $B^\omega \ni r \equiv p/q \in [0, 1) \subset \mathbb{R}$ is given as prefix in hexadecimal. k gives the first place with $s_k \neq r_k$. $\ominus$ indicates $s < r$, while $\oplus$ indicates $s > r$. $R = R(k-1)$; values (R) appear inside the WHILE loop.

Given that the sequence $\pi - 3 = .243F6A88\dots$ fails to meet the suggested approximations in places $s_1, s_2, s_6, s_9, s_{22-24}, s_{26-30}, s_{32}, s_{35} \dots$, we obtain

the discrepancy sequence $\mathbf{D}(0010.0110.0011.1111.0110.1010.1000.1000) = 1100.0100.1000.0000.0000.0111.0111.1101 = (d_k) = d$. The long 0-run between d_9 and d_{22} is of course the result of the large PD 292 and the excellent approximation $16/113$ (cf. $355/113$ for π). Also, the encoding $C(s) = v$ (compare with the $\ominus/\oplus$ sequence), is $C(s) = 00100.1110111.1.11111111000100100.1 \dots$.

7 Connections between rational and 2-adic complexities, and impulse response sequences

In the context of linear complexity, a periodic sequence $s = (s_1, s_2, \dots, s_T)^\omega$ of period T has linear complexity $T - \deg(\gcd(x^T - 1, S))$, where $S := \sum_{i=1}^T s_i x^{i-1} \in \mathbb{F}_2[x]$ assembles the digits of the period. For rational complexity, we similarly have:

Proposition 12 (Rational complexity of periodic sequences) *Let $s = (s_1, \dots, s_T)^\omega$ be a periodic sequence. Let $S := \sum_{i=1}^T s_i 2^{T-i} \in \mathbb{N}$, and let $g := \gcd(2^T - 1, S)$. Then s has rational complexity $R(s) = T - \lfloor \log_2(g) \rfloor$.*

Proof s can be generated as P/Q with $P := S$ and $Q := 2^T - 1$. Set $\tilde{P} := S/g$ and $\tilde{Q} := (2^T - 1)/g$. Then still $s = \tilde{P}/\tilde{Q} = (S/g)/((2^T - 1)/g) = S/(2^T - 1)$, which shows the upper bound $R(s) \leq \lceil \log_2((2^T - 1)/g) \rceil$. As $\mathbb{Z}$ is a unique factorization domain, there is no simpler way to represent $\tilde{P}$ and $\tilde{Q}$. Since $2 \nmid g$, $\log_2(g)$ is not integral, and thus $R(s) = \lceil \log_2((2^T - 1)/g) \rceil = T - \lfloor \log_2(g) \rfloor$. $\square$

In the following, $\hat{s} \in \mathbb{Q} \subset \mathbb{Z}_2$ refers to the 2-adic value of the ultimately periodic sequence $s \in B^\omega$. Note that $\hat{s} \neq s \in \mathbb{Q} \subset \mathbb{R}$ (reuse of symbol s in $\mathbb{R}_1$ and B^ω) in general.

Theorem 13 Connection between Rational and 2-adic Complexity

- (i) *Let $s = y^\omega$ be a purely periodic sequence. Then $R(s)$ equals the 2-adic complexity of the sequence with reversed period, $R(y^\omega) = A((\overleftarrow{y})^\omega)$.*
The symmetric 2-adic complexity [4], $A_{\text{SYM}}(y^\omega) := \min(A(y^\omega), A((\overleftarrow{y})^\omega))$ is equal to $\min(A(y^\omega), R(y^\omega)) = \min(R(y^\omega), R((\overleftarrow{y})^\omega))$.
- (ii) *Let $s = xy^\omega$, $x, y \in B^+$, be an ultimately periodic sequence. Then $R(s)$ and the 2-adic complexity $A((\overleftarrow{x})(\overleftarrow{y})^\omega)$ are both obtained from the numbers $(x)_2, (y)_2, 2^{|x|}, 2^{|y|} - 1$.*

Proof (i) For purely periodic sequences $s = y^\omega$, let $\overleftarrow{s} := (\overleftarrow{y})^\omega$. Then we have

$$s = \frac{(y)_2}{2^{|y|} - 1} \in \mathbb{R}, \quad \hat{s} = \frac{-(\overleftarrow{y})_2}{2^{|y|} - 1} \in \mathbb{Z}_2, \quad \overleftarrow{s} = \frac{(\overleftarrow{y})_2}{2^{|y|} - 1} \in \mathbb{R}, \quad \widehat{\overleftarrow{s}} = \frac{-(y)_2}{2^{|y|} - 1} \in \mathbb{Z}_2.$$

In this case, the reduced fractions $s = p/q$, $\overleftarrow{s} = m/n \in \mathbb{R}$, $\hat{s} = -m/n$, $\widehat{\overleftarrow{s}} = -p/q \in \mathbb{Z}_2$ satisfy $1 \leq p < q$ and $1 \leq m < n$, q, n odd, see also [7, Cor. 2.1]. Hence, the rational complexity—of purely periodic sequences—can indeed be computed via the FCSR apparatus from 2-adic complexity:

$R(y^\omega) = A((\overleftarrow{y})^\omega) = \lceil \log_2(q) \rceil = 1 + \lfloor \log_2(q) \rfloor$, if $T > 1$. The complexities also coincide for the special cases $R(0^\omega) = A(0^\omega) = 0$ and $R(1^\omega) = A(1^\omega) = 1$ (from $-1 \in \mathbb{Z}_2$).

The claim about the symmetric 2-adic complexity follows immediately.

- (ii) For ultimately periodic sequences $s = xy^\omega$ with $s' := \overleftarrow{x} (\overleftarrow{y})^\omega$, we similarly have

$$s = 2^{-|x|} \cdot \left((x)_2 + \frac{(y)_2}{2^{|y|} - 1} \right) \in \mathbb{R}, \hat{s} = (\overleftarrow{x})_2 + 2^{|x|} \cdot \frac{-(\overleftarrow{y})_2}{2^{|y|} - 1}, \hat{s}' = (x)_2 + 2^{|x|} \cdot \frac{-(y)_2}{2^{|y|} - 1} \in \mathbb{Z}_2,$$

where the first term of the sum corresponds to the preperiod, the second to the period, in all formulae. s and $\hat{s}'$ are built up from the same 4 parts, thus so are $R(s)$ and $A(s')$, but (of course) s and $\hat{s}'$ are not necessarily equal, neither are $R(s)$ and $A(s')$. $\square$

Example 14 Let $s = 1011(1100)^\omega$ and $t = 1101(0011)^\omega$,

with $(1011)_2 = 11$, $(\overleftarrow{1011})_2 = (1101)_2 = 13$, $(1100)_2 = 12$, $(\overleftarrow{1100})_2 = (0011)_2 = 3$

Then $s = \frac{11}{16} + \frac{1}{16} \cdot \frac{12}{15} = \frac{177}{240} = \frac{59}{80} \in \mathbb{R}$ and $\hat{s} = 13 + 16 \cdot \frac{-3}{15} = \frac{195-48}{15} = \frac{147}{15} = \frac{49}{5} \in \mathbb{Z}_2$.

We have $R(s) = \lceil \log_2(80) \rceil = 7$, and $A(s) = 1 + \lfloor \log_2(49) \rfloor = 6$.

Also, $t = \frac{13}{16} + \frac{1}{16} \cdot \frac{3}{15} = \frac{195+3}{240} = \frac{33}{40} \in \mathbb{R}$ and $\hat{t} = 11 + 16 \cdot \frac{-12}{15} = \frac{165-12}{15} = \frac{153}{15} = \frac{51}{5} \in \mathbb{Z}_2$.

We have $R(t) = \lceil \log_2(40) \rceil = 6$, and $A(t) = 1 + \lfloor \log_2(51) \rfloor = 6$.

$R(s) \neq A(t)$ are both obtained using the numbers 11, 12, 2^4 , $2^4 - 1$, while $R(t) = A(s)$ use the numbers 13, 3, 2^4 , $2^4 - 1$. The equality in the latter case is pure coincidence, $R(t)$ stemming from the denominator, $A(s)$ using the numerator.

Conjecture 15 Simulating R by A in $O(n^4)$

There is an $O(n^4)$ algorithm to compute the rational complexity without using the Binary CFE, applying the apparatus from 2-adic complexity—and vice versa.

Proof (Idea:) We can assume the separation into x and y after any position $0 \leq k \leq n$ within $s_1 \dots s_n$, and then apply Theorem 13(ii) to each case. This incurs another factor $n + 1$ in complexity. We thus obtain $n + 1$ rational approximations to s that are all valid on the given n -bit prefix. Among these approximations, the one with smallest Q defines the rational complexity.

Let the current approximation to $s = v\dots$ be $s \doteq (va\dots z)^\omega$.

We efficiently obtain z from v by first generating an FCSR for v , and then running the FCSR *backwards*. Thus, we need $O(|v|)$ steps instead of the full period, which might be considerably larger.

Incrementally computing the best rational approximation to a sequence $s = v\bar{a}\dots$ from the approximation to the shorter prefix $s = v\dots$ via 2-adic complexity / FCSRs is not straightforward:

If the next bit after v in s happens to be a , the discrepancy is zero and the approximation is maintained. If however, s continues with $\bar{a} = 1 - a$, then $s = v\bar{a}\dots$ will have a best approximation $s \doteq v_1(v_2\bar{a}\dots\tilde{z})^\omega$, where firstly v possibly will be split into a preperiod $v_1 \neq \varepsilon$ and the start v_2 of the new period, and secondly the end of the new period, $\tilde{z}$, has no direct relation with z . Also, the period length may grow or shrink considerably.

If the best approximation to v was already ultimately periodic with preperiod, things are similar. The length of the preperiod may change: $s = uv\dots \doteq uv^\omega$ may lead to $s = uvuv\dots \doteq (uv)^\omega$ for one extension, and to $s = uv10^{|uv|+1}\dots \doteq uv10^\omega$ in another case. In other words, the new preperiod may be empty or include the whole previous prefix, or anything in between.

Since the new periods terminate in different words, $\tilde{z} \neq z$, for each successive approximation—unless the previous value remains the same—we have to restart the FCSR

from time step 1, a further factor n . Hence, we get a new approximation for a given separation position in $O(n^3)$. All in all, we can simulate rational approximation by 2-adic approximation in $O(n^4)$.

The four factors n correspond to (i) the operand lengths, (ii) the assumed length of the preperiod, i.e. the position of the separation, (iii) the restart time for the FCSRs with the reversed end of the new period, $\overleftarrow{z}$, and (iv) the growing prefix size of s .

As far as we can see, speed-up via FFT techniques is not feasible due to (iii), since every two steps on average (for i.i.d. sequences) we have to restart the FCSRs with the new value $\overleftarrow{z}$ instead of the $\overleftarrow{z}$ (Open Problem 7). $\square$

Theorem 16 Periods of impulse response sequences

- (i) Given an odd $q \in \mathbb{N}$, let its impulse response sequence $s \in B^\omega$ be the binary expansion of $1/q$. Then the period of s divides $\varphi(q)$.
- (ii) The period of a/q is at most that of the impulse response sequence $1/q$, for any a .
- (iii) For $q = 2^n - 1$, the period of the associated sequence $1/q$ divides n . Hence, unlike the LFSR case, there are no sequences with period $2^n - 1$ and rational complexity n .
- (iv) Let p be a prime and 2 a generator of the multiplicative group $\mathbb{F}_p^*$. Then a/p has period $p - 1$ for all $a \neq 0$.
- (v) For $n \in \mathbb{N}$, $N := 2^n$, we can expect a prime p with $\langle 2 \rangle = \mathbb{F}_p^*$ within the range $[N - 2n, N - 1]$ (where $N - 1$ is excluded by (iii)).

Proof Let $\text{ord}_q(g)$ be the order of the (multiplicative) cyclic subgroup $\langle g \rangle \subset (\mathbb{Z}/q\mathbb{Z})^*$. For prime $q = p$ we also write $\langle g \rangle \subset \mathbb{F}_p^*$. Our case is $g = 2$ for binary shifts.

- (iii) The residues mod q of $1/q$ are $\{2^k \pmod{q} \mid k = 1, 2, \dots, \text{ord}_q(2)\} = \langle 2 \rangle \subset (\mathbb{Z}/q\mathbb{Z})^*$, the group of units mod q . Since $|(\mathbb{Z}/q\mathbb{Z})^*| = \varphi(q)$, the claim now follows from Lagrange's (subgroup index) theorem.
- (ii) The residues $a \cdot 2^k \pmod{q}$ for a/q differ from those for $1/q$ by a factor a . In particular they repeat after the same period, possibly having a shorter least period.
- (iii) $2^n = 1 \pmod{q} \iff \text{ord}_q(2) \mid n$.
- (iv) Since 2 generates $\mathbb{F}_p^*$, i.e. 2^k passes through all nonzero residues mod p , the first $p - 1$ shifts of $1/p$ pass through all a/p with $a \neq 0$. Thus they all have the period $p - 1$.
- (v) By the Prime Number Theorem, the probability of being prime for some p near N is $1/\log_e(N) \doteq 1/(0.69 \cdot n)$. Assuming Artin's conjecture (on primitive roots) [12] for prime p , 2 generates $\mathbb{F}_p^*$ with probability $C_{\text{Artin}} = \prod_p \left(1 - \frac{1}{p(p-1)}\right) \doteq 0.373955$. The joint probability $C_{\text{Artin}} \cdot 1/(0.69 \cdot n) > 1/2n$ yields the result.

$\square$

8 Comparison of R with A and L

Rational complexity R , linear complexity L , and 2-adic complexity A all assign a finite complexity to ultimately periodic sequences. However, the three complexities assign low values to quite different sequences / prefixes [20].

In Table 7, we give the number of 30-bit prefixes which have an unremarkable value in one of the complexities ($L \geq 14$, $A \geq 14$, and $R \geq 15$, respectively), being higher than or at most one smaller than the expected value. We then classify these prefixes according to

Table 7 Number of sequences with one high complexity value, for L , A , and R at $n = 30$

$L \geq 14$				$A \geq 14$			
$A :$	R: 0..10	11..15	16..30	$L :$	R: 0..10	11..15	16..30
0..9	252	64425	123802	0..9	16	51795	107990
10..14	65256	65564743	149388020	10..14	49901	53639684	122678310
15..30	228863	255061144	592060508	15..30	240626	266201775	617172734
$R \geq 15$							
$A :$	$L : 0..9$	10..14	15..30				
0..9	14	32421	150405				
10..14	31732	35724773	177284599				
15..30	125736	139456530	700529908				

the remaining two complexities into 3 classes: expected value or higher (third column/row), off by at most $5 \approx \log_2(30)$ (middle), and the definitely atypical case, off from the expected value by more than a logarithmic difference (first column/row).

In particular, we have in boldface the numbers of sequences, which are *only* recognized as low-complexity by rational complexity, but neither by linear, nor 2-adic complexity.

Over the set of prefixes of length 30, the three complexities L , A , and R are essentially uncorrelated, both in pairs and all three combined. Very low (0 or 1) and very high ($30 = n$) complexities correspond to the same sequences for L , A , and R , namely $(0^{30}; 1^{30}, 10^{29}; 0^{29}1)$. For most sequence prefixes, however, the three complexities behave statistically unrelated. We examine prefixes of length $n = 30$ in the following lemma.

Lemma 17 Low-Correlation-Lemma

Let $\#(L = l, A = a) = |\{s \in B^{30} : L(s, 30) = l \wedge A(s, 30) = a\}|$ etc.

Define the “fraction of non-overlapping of distributions” as

$$\rho_{LAR} = \sum_{l=0}^{30} \sum_{a=0}^{30} \sum_{r=0}^{30} \left| \frac{\#(L = l, A = a, R = r)}{2^{30}} - \frac{\#(L = l)}{2^{30}} \cdot \frac{\#(A = a)}{2^{30}} \cdot \frac{\#(R = r)}{2^{30}} \right|,$$

$$\rho_{LA} = \sum_{l=0}^{30} \sum_{a=0}^{30} \left| \frac{\#(L = l, A = a)}{2^{30}} - \frac{\#(L = l)}{2^{30}} \cdot \frac{\#(A = a)}{2^{30}} \right|, \quad \rho_{LR}, \rho_{AR} \text{ analogously.}$$

Then $\rho_{LAR} \doteq 0.00938$, $\rho_{LA} \doteq 0.00494$, $\rho_{AR} \doteq 0.00720$, $\rho_{LR} \doteq 0.00528$.

Proof By numerical evaluation. More than 99% of all sequence prefixes behave according to the distribution expected for completely uncorrelated complexity measures, for all 4 correlations. $\square$

9 Deviation from isometric behaviour

For the linear complexity, the encoding of PDs and the induced isometry, the discrepancy sequence, coincide. Both are given by $\mathbf{K}(s)$. In our case, the encoding $C(s)$ and the discrepancy operator $\mathbf{D}(s)$, like $\mathbf{K}$ an isometry on B^ω , are different. This section investigates, by how much.

Table 8 Length extension γ_b for fixed partial denominator and CFE for ϕ_1, ϕ_4

b	ϕ_b	L	l	γ_b	γ_b [$\pm\%$]	$1/\gamma_b$ [$\pm\%$]	b	ϕ_b	L	l	γ_b	γ_b [$\pm\%$]	$1/\gamma_b$ [$\pm\%$]
1	1.618	1.39	1	0.72	−28.0	+38.8	7	7.140	5.67	5	0.88	−11.8	+13.4
2	2.414	2.54	3	1.18	+18.0	−15.2	8	8.123	6.04	7	1.16	+15.8	−13.6
3	3.303	3.45	3	0.87	−13.0	+14.9	15	15.066	7.82	7	0.89	−10.6	+11.8
4	4.236	4.17	5	1.20	+20.0	−16.7	16	16.062	8.01	9	1.12	+12.3	−10.9
5	5.193	4.75	5	1.05	+5.21	−4.94	31	31.032	9.91	9	0.91	−9.2	+10.1
6	6.162	5.25	5	0.95	−4.67	+4.94	32	32.031	10.00	11	1.10	+10.0	−9.07

i	−1	0	1	2	3	4	5	6	7	8	9	
P_i	1	0	1	1	2	3	5	8	13	21	34	yields ϕ_1
Q_i	0	1	1	2	3	5	8	13	21	34	55	
P_i	1	0	1	4	17	72	305	1292	...			yields ϕ_4
Q_i	0	1	4	17	72	305	1292	5473	...			

$\phi_1: \frac{l}{L} = \frac{(9 \times 1)}{\log_2(55^2)} = 0.778 \dots$ $\phi_4: \frac{l}{L} = \frac{(6 \times 5)}{\log_2(5473^2)} = 1.208 \dots \approx \gamma_4$

The last step for ϕ_1 yields $(9 - 8)/(\log_2(55^2) - \log_2(34^2)) = 0.72 \dots \approx \gamma_1$

Lemma 18 Information Content of Prefixes from $s, d = \mathbf{D}(s)$, and $C(s)$

Let $s \in \mathbb{R}_1$ satisfy the Gauß-Kuz'min measure μ_{GK} . For a given prefix length $n \in \mathbb{N}$, let $r = P_{2i}/Q_{2i} = [b_1, \dots, b_{2i}]$ be the first approximation that matches the first n bits of s , i.e. $r_k = s_k, 1 \leq k \leq n$.

Then asymptotically, encoding length and precision satisfy $\lim_{n \rightarrow \infty} \frac{\sum_{j=1}^{2i} l_{I,II}(b_j)}{n} \doteq 1.024$.

Proof From $\sum_{b \in \mathbb{N}} l_{I,II}(b) \cdot \mu_{GK}(b) \doteq 3.50698$, Theorem 1(iii), and $\frac{3.50698}{3.42371} \doteq 1.024$. $\square$

Lemma 18 shows that for numbers satisfying the Gauß-Kuz'min measure, the code length $C(s)$ grows faster than the considered prefix length of the sequences s and $d = \mathbf{D}(s)$ by a factor of (only) 1.024. One could say that we miss our goal to mimic the isometric behaviour of $\mathbf{K}$ by just 2.4% (on average).

For numbers not satisfying the Gauß-Kuz'min measure, e.g. rationals or quadratic-algebraic numbers, the code may grow faster or slower. We compare the coding length l for $\Phi_b := [b, b, b, b, \dots] = (\sqrt{b^2 + 4} - b)/2$ with its gain in precision. In this case, the convergents' denominators (and numerators) asymptotically increase by a factor

$\phi_b := (b + \sqrt{b^2 + 4})/2 (= b + \Phi_b)$, from $\frac{\begin{array}{|c|} b_i = b \end{array}}{1|\phi_b|b \cdot \phi_b + 1}$ (Perron's schema) with

$b \cdot \phi_b + 1 \stackrel{!}{=} \phi_b^2 \iff \phi_b^2 - b\phi_b - 1 = 0$.

As $|r - P_i/Q_i|$ decreases like $\Theta(Q_i^{-2}) = \Theta(\phi_b^{-2i})$ (since $r \notin \mathbb{Q}$), see [14, Satz 2.10], each additional PD b asymptotically yields $L := \log_2(\phi_b^2)$ more bits with $r_k = s_k$.

Since b is encoded by $l = l_{I,II}(b) = 1 + 2\lfloor \log_2(b) \rfloor$ bits, we obtain the length extension for the encoding by

$$\gamma_b := \frac{l}{L} = \frac{1 + 2\lfloor \log_2(b) \rfloor}{2 \cdot \log_2(\phi_b)}.$$

The values γ_b decrease monotonically with b , from $b = 2^t$ to $b = 2^{t+1} - 1$ with fixed $l = 2t + 1$, then jump to a local maximum at 2^{t+1} etc. From Table 8, we can infer that $\gamma_1 = 0.72$ is the minimum with a saving of 28% in coding size, 1000 bits of precision require only 720 bits of $C(s)$, while $\gamma_4 = 1.20$ is the maximum value for a length extension of 20%, the 1000 bits $s_1, \dots, s_{1000}$ require about 1200 bits of $C(s)$. Hence, ϕ_1 has the shortest, ϕ_4 the longest code, compared with attained precision. Comparing with Lemma 18, the value 1.024 has to be replaced by the range $[0.72, 1.20]$ to cover all $s \in \mathbb{R}_1 \setminus \mathbb{Q}_1$.

For $s \in \mathbb{Q}_1$, we have $l/L \rightarrow 0$ asymptotically, a finite CFE yields infinite precision.

10 The Q Tree: an ultrametric ordering of $\mathbb{Q}_1 = \mathbb{Q} \cap [0, 1]$

In this section, we generate the Q Tree (Table 9), an isometric version of the V_{10} Tree. Its upper k levels, plus the allzero word, include all prefixes from B^k exactly once. We show, how to use the Q Tree to accelerate the Binary CFE.

First define an infinite binary tree with labels, the *Expanded Q Tree*: We read out the V_{10} Tree [22] in breadth-first order, prepending 0/1, and put each value P/Q as label on every node whose address is a prefix of $s = P/Q$ and that has not yet been labeled.

Example: The label $s = 0/1$ from 0^ω is put on nodes ε (the root), 0, 00, 000 etc. Then $s = 1/2 = 10^\omega$ labels 1, 10, 100, etc., while ε is already taken. $s = 1/4 = 010^\omega$ skips ε , 0 and labels from 01 onward. $s = 2/3 = (10)^\omega$ skips ε , 1, 10 and labels 101, 1010 and so on.

Theorem 19 *Nodes and labels of the Expanded Q Tree*

- (i) *The labels $P/Q = s$ of the 2^k nodes on level k yield each prefix $v \in B^k$ exactly once.*
- (ii) *Exactly one of the two child nodes has the same label as the parent node.*

Proof (i) By construction: Any prefix $v \in B^*$ is met just at the node with address v .

- (ii) Once a node receives a label s , the same s moves further down the addresses and labels a —yet unlabeled—child node of each visited node. □

To generate the *Q Tree* from the Expanded Q Tree, for each node replace its label by the one of its child nodes, which is different. This in particular removes all labels 0/1.

Theorem 20 *Equidistribution of Prefixes in the Q Tree*

- (i) *The binary expansions of the labels of the $2^k - 1$ nodes on levels $1 \dots k - 1$ of the Q Tree, plus the special value 0^k , include every prefix from B^k exactly once.*
- (ii) *In the Q Tree, each word $v \in B^k \setminus \{0^k\}$ appears as prefix of the binary expansion of the label P/Q , at some point on the path from the root to the node with address $v_1 \dots v_{|v|-1}$.*

Proof (i) By construction and Theorem 19. All entries on level k from the Expanded Q Tree are preserved somewhere further up—except 0^ω .

- (ii) The prefix was present at address v in the Expanded Q Tree and has moved up by construction. However, all such places, where it can move are just those on the path from the root to the parent of the node with address v , which has address $v_1 \dots v_{|v|-1}$. □

Table 9 Address $v \in B^*$, Q Tree ($Q[v]$), and V_{10} Tree ($C^{-1}(v10^\omega)$)

Corollary 21 *Writing out the Q Tree, preceded by the value 0, yields an ultrametric ordering of $\mathbb{Q}_1$ (where the first 2^k entries of the sequence include all prefixes of length k exactly once, for all $k \in \mathbb{N}$) : $0/1, 1/2, 1/4, 4/5, 1/8, 2/5 \dots$.*

The procedure of Table 4 (left part) is simplified considerably, if the Q Tree is known, see Table 10. Note that now the WHILE loop runs through r , s , and d . The original WHILE loop has been replaced by the single access to $Q[v]$, whenever the current approximation does no longer hold, and at the exact address v for the new approximation P/Q . However, the Q Tree is as yet only a by-product after setting up the V_{10} Tree, see Open Problem 6.

11 Binary CFE III: states, incremental change of b_i, P_i, Q_i

Recalling Section 4, we start with the full cylinder set B^ω or $\mathbb{R}_1$ as possible encodings. Using the known correct prefix v (initially we know nothing, $v = \varepsilon$), we successively cut the cylinder set into half by probing with the trial value $v10^\omega \xrightarrow{C^{-1}} r$, which sits at the center

Table 10 Binary CFE using Q Tree

```
1.  $r := 0^\omega$ ;  $i := 1$ 
2. FOREVER
3.   WHILE ( $r_i = s_i$ ) BEGIN  $d_i := 0$ ;  $i++$  END
4.    $v = (r_1 \dots r_{i-1})$ ;  $d_i := 1$ ;  $i++$ 
5.    $r = FQSR(Q[v])$ 
// consult Q Tree at  $v$ , label is new  $P/Q$  for  $FQSR$ 
6. ENDFOR
```

Table 11 Encodings and states

	IN: $v.10^\omega$	OUT: $\ominus : v0.10^\omega$, for $s < r$, $v_k = 0$ OUT: $\oplus : v1.10^\omega$, for $s > r$, $v_k = 1$	Next- state
A	$.1 0 \equiv [1, 1]$	$0.10 0 \equiv [3, 1]$ $1 .100 \equiv [1, 2]$	B_1 $\bar{A}$
B_j	$0^j.10^j 0 \equiv [2^{j+1} - 1, 1]$	$0^j0.10^{j+1} 0 \equiv [2^{j+2} - 1, 1]$ $0^j1.10^{j-1} 0 \equiv [2^j + 2^{j-1} - 1, 1]$	B_{j+1} $C_j, m := j$
C_j	$0^m 1 *^{m-j}.10^{j-1} 0 \equiv [N, 1]$	$0^m 1 *^{m-j} 0.10^{j-2} 0 \equiv [N + 2^{j-2}, 1]$ $0^m 1 *^{m-j} 1.10^{j-2} 0 \equiv [N - 2^{j-2}, 1]$	C_{j-1} C_{j-1}
$j \geq 2$			
D	$0^m 1 *^{m-1}.1 0 \equiv [N, 1]$	$0^m 1 *^{m-1} 0 .100 \equiv [N + 1, 2]$ $0^m 1 *^{m-1} 1 .100 \equiv [N, 2]$	$\bar{A}$ $\bar{A}$
$= C_1$			
$\bar{A}$	$\dots .100 \equiv [\dots, 2]$	$\dots 0 .1 0 \equiv [\dots, 1, 1, 1]$ $\dots .1.1000 \equiv [\dots, 4]$	A $\bar{B}_1$
$\bar{B}_j$	$\dots 1^j.10^{j+2} \equiv [\dots, 2^{j+1}]$	$\dots 1^j0.10^{j-1} \equiv [\dots, 2^j + 2^{j-1}]$ $\dots 1^j1.10^{j+3} \equiv [\dots, 2^{j+2}]$	$\bar{C}_j, m := j$ $\bar{B}_{j+1}$
$\bar{C}_j$	$\dots 1^m 0 *^{m-j}.10^{j-1} \equiv [\dots, N]$	$\dots 1^m 0 *^{m-j} 0.10^{j-2} \equiv [\dots, N - 2^{j-2}]$ $\dots 1^m 0 *^{m-j} 1.10^{j-2} \equiv [\dots, N + 2^{j-2}]$	$\bar{C}_{j-1}$ $\bar{C}_{j-1}$
$j \geq 2$			
$\bar{D}$	$\dots 1^m 0 *^{m-1}.1 \equiv [\dots, N]$	$\dots 1^m 0 *^{m-1} 0 .1 0 \equiv [\dots, N - 1, 1, 1]$ $\dots 1^m 0 *^{m-1} 1 .1 0 \equiv [\dots, N, 1, 1]$	A A
$= \bar{C}_1$			

of the current cylinder set (we could as well have used $v01^\omega$, yielding the same r , with convention (iv) for the ambiguity). This binary search leads to a new approximation:

$$v \mapsto v10^\omega = C_I(b_1)|C_{II}(b_2)|\dots|C_{II}(b_{2l})|C_I(\aleph_0) \mapsto [b_1, b_2, \dots, b_{2l}] = \frac{P}{Q} = r = C^{-1}(v10^\omega).$$

In order to obtain an efficient implementation of this part, we maintain the state of a Finite State Machine plus Counter (FSM+C) and the current as well as an auxiliary convergent, P/Q and P'/Q' , with initial values $1/2$ and $0/1$, respectively. The state set is $\{A, \dots, D, \bar{A}, \dots, \bar{D}\}$, the counter appears as index j for states $B, C, \bar{B}, \bar{C}$. We start in state A .

The states $A \dots D$ handle PDs with odd index $2i + 1$, while states $\bar{A} \dots \bar{D}$ for PDs with even index operate exactly 0-1-inverse to $A \dots D$. Since we always maintain an even number of

Table 12 Update of convergents

State	PD		$\ominus, v_k = 0$		$\oplus, v_k = 1$	
	index	$P'^+ :=$	$P^+ := \dots$	$P'^+ :=$	$P^+ := \dots$	
A	$2i + 1$	P'	$P + (P' << 1)$	$P - P'$	$(P << 1) - P'$	
B_j	$2i + 1$	P'	$P + (P' << (j + 1))$	P'	$P - (P' << (j - 1))$	
C_j	$2i + 1$	P'	$P + (P' << (j - 2))$	P'	$P - (P' << (j - 2))$	
D	$2i + 1$	P	$(P << 1) + P'$	$P - P'$	$(P << 1) - P'$	
$\bar{A}$	$2i + 2$	$P - P'$	$(P << 1) - P'$	P'	$P + (P' << 1)$	
$\bar{B}_j$	$2i + 2$	P'	$P - (P' << (j - 1))$	P'	$P + (P' << (j + 1))$	
$\bar{C}_j$	$2i + 2$	P'	$P - (P' << (j - 2))$	P'	$P + (P' << (j - 2))$	
$\bar{D}$	$2i + 2$	$P - P'$	$(P << 1) - P'$	P	$(P << 1) + P'$	

PDs, for an odd current index the PD value b_{2l+1} is distributed as $[\dots, b_{2l+1} - 1, 1]$. This is the only difference between the two groups of states, apart from the bit complement. The update of the state and of the two convergents are shown in Tables 11 and 12 (for Q, Q' analogously). “ $<<$ ” is the left shift. The first $2i$ PDs are fixed and omitted.

Observe that no multiplication (as in $P_i := b_i \cdot P_{i-1} + P_{i-2}$, $Q_i := b_i \cdot Q_{i-1} + Q_{i-2}$) is necessary. It is replaced by an incremental shift-and-add. We thus obtain property (INC)—everything is done bitwise, incrementally, and only additions and shifts are used (the shifts by arbitrarily large j can be avoided by shifting P', Q' in each state $B_j, C_j, \overline{B_j}, \overline{C_j}$).

12 Feedback in $\mathbb{Q}$ shift registers FQSRs: Reduce-By-Feedback

We have to obtain the rational sequence $r = P/Q$ to be compared with s . Paper-and-pencil long division in base 2 of P by Q (here $P < Q$ is assumed) proceeds as in Table 13 (left).

An FQSR is thus just a standard long accumulator X for values from $\mathbb{N}_0$ with a shift-and-add unit. It requires a full length compare in each cycle. The next part shows, how to replace that time-consuming compare by a 2-bit compare and a different way of keeping X (almost) within the range $[0, Q)$.

Remark 22 In analogy with the LFSR case, we can consider FCSRs as “Fibonacci style” and FQSRs as “Galois style” implementation of feedback with carry.

If implementing in software, skip directly to Theorem 24.

An alternative implementation of division in $\mathbb{Q}$ and thus FQSRs, by Reduce-By-Feedback (see [21]), is shown in Table 13, right hand side.

Let $n := \lceil \log_2(Q) \rceil$, $N := 2^n$, hence $N/2 < Q \leq N$. We use an $(n+2)$ -bit register for X , initially filled with P . Setting $K \equiv N \pmod{Q}$ that is $K := N - Q$, we may replace $X := X - Q$ by $X := X - N + K$.

Let m denote the two leftmost bits of the FQSR, of content 0,1,2, or 3. We replace the decision ($X \geq Q?$), a long compare, by the two-bit-check ($X \geq N?$) that is ($m \geq 1?$).

If necessary, for $m \neq 0$, we subtract $m \cdot N$ (by setting $m := 0$) and compensate by adding the constant $m \cdot K$. In this way, the contents of X is always correct mod Q . It may, however, exceed $Q - 1$, lying in the range $Q \leq X < N$.

We now show that $m \leq 2$ is always guaranteed.

Table 13 Algorithms: Division in $\mathbb{Q}$, Reduce-By-Feedback

$X := P$	$X := P // m = 0$
FOREVER	FOREVER
$X := 2 \cdot X$	$X := (X << 1) + m \cdot K - m \cdot N$
IF $X \geq Q$ THEN Output 1; $X := X - Q$	Output m
ELSE Output 0 ENDIF	ENDFOR
ENDFOR	

Proposition 23 Overflow-freeness of the range $m \in \{0, 1, 2\}$

Let $0 \leq m \leq 2$. Then after one clock cycle we still have $0 \leq m^+ = \lfloor (2X + mK)/N \rfloor \leq 2$.

Proof We have $K = N - Q$ and $N/2 < Q \leq N$. Therefore, $0 \leq K < N/2$ and $0 \leq 2X + 0 \cdot K \leq 2X + m \cdot K \leq 2X + 2 \cdot K < 2 \cdot N + 2 \cdot N/2 = 3N$, which assures that the range $\{0, 1, 2\}$ for m is also satisfied in the next step. $\square$

We want the sequence $r = (r_k) \in \{0, 1\}^\omega$, but instead we obtain $(m_l) \in \{0, 1, 2\}^\omega$. While we have $\sum_{k \in \mathbb{N}} r_k 2^{-k} = \sum_{l \in \mathbb{N}} m_l 2^{-l}$ by design, we must convert (m_l) into (r_k) :

```

ctr := 0; k := 0
FOR l = 1, 2, ...
  IF m_l = 0 THEN Output (r_k = 0, r_{k+1} = ... r_{k+ctr} = 1); k := k + ctr + 1; ctr := 0 ENDIF
  IF m_l = 1 THEN ctr := ctr + 1 ENDIF
  IF m_l = 2 THEN Output(r_k = 1, r_{k+1} = ... r_{k+ctr} = 0); k := k + ctr + 1; ctr := 0 ENDIF
ENDFOR

```

Essentially, a run of ones with leading zero is saved for later and output when $m = 0$. With $m = 2$, a carry ripples through the 1-run, converting it to 10^{ctr} . We start anew with a leading zero, not yet output. The value r_0 will always be zero and should be thrown away (since $P < Q$, we do never start with $m = 1^k 2 \dots$).

More on the implementation of FQSRs in hardware can be found in [21], including the use of Brickell's delayed-carry-adder technique: Paper-and-pencil long division is a subset of RSA via modular exponentiation $\supseteq$ multiplication mod $Q \supseteq$ shift-and-add mod $Q \supseteq$ shift (no add) mod Q = long division.

Theorem 24 Fast Reinitialisation of FQSRs

We denote values at timestep $k \in \mathbb{N}$ by the superscript $^{(k)}$.

Consider 3 FQSRs with initial contents $P^{(1)}$, $P'^{(1)}$, and $P''^{(1)} := P^{(1)} + bP'^{(1)}$ for some $b \in \mathbb{N}$, with feedback values Q , Q' , and $Q'' := Q + bQ'$, and where N , N' , N'' are the respective powers of 2 corresponding to the sizes of Q , Q' , Q'' .

Let $m^{(k)} := 2 \cdot P^{(k)} \bmod Q \in \{0, 1\}$, and similarly $m'^{(k)}$, $m''^{(k)}$ be the respective feedback bits and $P^{(k+1)} := 2P^{(k)} - m^{(k)}Q$, $P'^{(k+1)} := 2P'^{(k)} - m'^{(k)}Q'$, $P''^{(k+1)} := 2P''^{(k)} - m''^{(k)}Q''$ be the register contents in the next timestep. Then:

(i) If $m^{(k)} = m'^{(k)}$ and the weighted sum property $P''^{(k)} = P^{(k)} + bP'^{(k)}$ holds, then also $m''^{(k)} = m^{(k)} (= m'^{(k)})$ and the property remains valid for $k + 1$, $P''^{(k+1)} = P^{(k+1)} + bP'^{(k+1)}$.

(ii) Let $r := P^{(1)}/Q^{(1)}$ and $r' := P'^{(1)}/Q'^{(1)}$ coincide for the first n bits, $r_k = r'_k$, $1 \leq k \leq n$. Then $r'' := P''^{(1)}/Q''^{(1)}$ also satisfies $r''_k = r_k$, $1 \leq k \leq n$, and the contents of the 3 FQSRs generating r , r' , and r'' satisfy the weighted sum property $P''^{(k)} = P^{(k)} + b \cdot P'^{(k)}$ for all timesteps $1 \leq k \leq n$.

Proof (i) First assume $m = m' = 0$. Then $P < Q$, $P' < Q'$ and therefore $P'' = P + bP' < Q + bQ' = Q''$, hence $m'' = 0$. For $m = m' = 1$, analogously $P \geq Q$, $P' \geq Q'$ and therefore $P'' = P + bP' \geq Q + bQ' = Q''$, hence $m'' = 1$.

Also, $P''^{(k+1)} = 2P''^{(k)} - m''^{(k)}Q'' = 2(P^{(k)} + bP'^{(k)}) - (m^{(k)}Q + m'^{(k)}bQ') = (2P^{(k)} - m^{(k)}Q) + b(P'^{(k)} - m'^{(k)}Q') = P^{(k+1)} + b \cdot P'^{(k+1)}$.

(ii) By induction on k with (i), using $r_k = m^{(k)}$, $r'_k = m'^{(k)}$, and $r''_k = m''^{(k)}$. $\square$

Remark 25 By Theorem 24, we may maintain two registers for the current and the previous convergent, P/Q and P'/Q' . Whenever a new FQSR with P^+, Q^+ is required, we set its current contents from P, P' by the weighted sum property, and may then drop P', Q' . We only have to repeat the (few) clock cycles starting from the last mismatch. Overall, we advance every contents $P/Q, P'/Q',$ or P^+/Q^+ from one mismatch to the one after the next, a total of $2n$ clock cycles or $O(n)$.

Theorem 26 Bit Complexity

The bit complexity of computing the current best approximations P/Q , the discrepancy sequence $\mathbf{D}(s)$, the coding sequence $C(s)$, and the rational complexity profile corresponding to the prefix $s_1, \dots, s_n$ of length n is $O(n^2)$.

Proof Calculating P/Q requires a factor n for the growing wordlength and an amortized number of $2n$ cycles according to the previous theorem and remark, a total of $O(n^2)$ bit operations. $\mathbf{D}(s) = (d_k)$ and $C(s)$ are obtained “on the fly” in the algorithm from Table 4 (left), in $O(1)$ operations per cycle or a total of $O(n)$. Observe that the comparison $(s_1 \dots s_k) \leq (r_1 \dots r_k)$ can effectively be done on a suffix, whose start index is maintained and updated separately for $\ominus$ and $\oplus$: We always have “ $r[\ominus] < s < r[\oplus]$ ”, the r values from Cases $\ominus$ are strictly increasing towards s from below, the values for Cases $\oplus$ strictly decrease from above, and thus once a bit position matches the corresponding s_k , this fact will be maintained. Hence, for any index k , the corresponding s_k is matched exactly twice, once in a Case $\ominus$, once for a Case $\oplus$, a total of $O(n)$. The rational complexity profile is obtained by comparing the length of the current Q against $R(s, k-1)$. Since $R(s, n) \leq n$, the comparison is $O(n)$ per step, $O(n^2)$ in total. $\square$

Remark 27 This coincides with the complexity of computing L or A without using FFT techniques, $O(n^2)$. However, there are FFT implementations both for linear (Blackburn [2]) and 2-adic complexity (Klapper [8, p. 137]) that need only $O(n \log(n)^2 \log \log(n))$ bit operations. See also Niederreiter and Vielhaber [13] for an algorithm that computes *all* discrepancy sequences of all shifted sequences $(\mathbf{K}(s_k, \dots, s_n)_i, 1 \leq i \leq n-k+1, 1 \leq k \leq n)$ in overall time $O(n^2)$, i.e. in a constant number of bit operations per output bit.

13 Open problems

1. When the approximation is very good, e.g. after n consecutive discrepancy bits $d_k = 0$, can we extend v directly by, say, $01^{[0.72n]}$ or $10^{[0.72n]}$, using the results in Section 9?
2. How, if at all, can the binary procedure be adapted to sequences over $\mathbb{F}_p, p > 2$?
3. Can we use FFT techniques to achieve an overall bit complexity of $O(n \log(n)^2 \log \log(n))$ for the rational complexity, like in the case of L and A ?
4. Give closed formulae for the rational complexity values $N_R(n, c)$, $E(R(n))$ and $\text{Var}(R(n))$, for even and odd n . Describe the numbers $v_E^a \approx -0.06$ and v_V^a from Conjecture 7.
5. How does the rational complexity behave for sequences with perfect autocorrelation, in particular compared with linear complexity and 2-adic complexity?
6. Can we construct the Q Tree without previously constructing the V_{10} Tree via the FSM+C of Section 11? (this would also solve Problem 1)
7. Prove Conjecture 15. Can we speed up the simulation of rational complexity by 2-adic complexity (or vice versa) below the suggested $O(n^4)$ complexity?

8. R is a CFE on the Archimedean $\mathbb{R}$, while A is an isometry on $\mathbb{Z}_2$ (ultrametric, no CFE). Can we reconcile R with A beyond the case of ultimately periodic sequences?

14 Conclusion

We have introduced *Rational Complexity* R as a novel complexity measure for binary sequences, a third way to assess the pseudorandomness of binary sequences by comparison with ultimately periodic sequences, besides linear and 2-adic complexity, L and A .

We presented the *Binary CFE Algorithm*, a pseudo-ultrametric, bit-by-bit version of the standard continued fraction expansion in $\mathbb{R}$, making use of *Feedback in $\mathbb{Q}$ Shift Registers*. The algorithm avoids the costly inversion $s^{(i)} := \{s^{(i-1)}\}^{-1}$ in $\mathbb{R}$.

We also defined two prefixfree complete codes, C_I and C_{II} , mapping the partial denominators from $\mathbb{N}$ to $\{0, 1\}^*$ and following the Gauß-Kuz'min measure. C_I and C_{II} yield a monotonic function $C: \mathbb{R}_1 \rightarrow B^\omega$, mapping reals to CFE encodings. We have seen that R , L and A differ pairwise on considerable parts of the prefix space B^{30} and thus all three measures should be used to examine a given sequence for pseudo-randomness.

Acknowledgements We sincerely thank the anonymous reviewers, both at SETA 2020 and *Cryptography and Communications* for their valuable and detailed suggestions.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Berlekamp, E.: Non-binary BCH decoding. TR North Carolina State University Department of Statistics (1966)
2. Blackburn, S.: Fast rational interpolation, Reed-Solomon decoding, and the linear complexity profiles of sequences. *IEEE Trans. IT* **43**(2), 537–548 (1997)
3. Dornstetter, J.L.: On the equivalence between Berlekamp's and Euclid's algorithm. *IEEE Trans. IT* **33**(3), 428–431 (1987)
4. Hu, H., Feng, D.: On the 2-adic complexity and the k -error 2-adic complexity of periodic binary sequences. In: SETA 2004 LNCS, vol. 3486, pp. 185–197 (2005)
5. Khintchine, A.: Metrische kettenbruchprobleme. *Comput. Math.* **1**, 361–382 (1935)
6. Klapper, A., Goresky, M.: 2-Adic shift registers. In: *Crypto '95 LNCS*, vol. 963, pp. 262–273 (1995)
7. Klapper, A., Goresky, M.: Cryptanalysis based on 2-Adic rational approximation. In: *Crypto '95, LNCS*, vol. 963, pp. 262–273 (1995)
8. Klapper, A., Goresky, M.: Feedback shift registers, 2-adic span, and combiners with memory. *J. Crypt.* **10**, 111–147 (1997)
9. Kuzmin, R.O.: Sur une problème de Gauss. *Atti Congr. Int. Bologna* **6**, 83–89 (1928)
10. Lévy, P.: Sur les lois de probabilité dont dépendent les quotients complets et incomplets d'une fraction continue. *Bull. S.M.F.* **57**, 178–194 (1929)
11. Massey, J.: Shift-register synthesis and BCH decoding. *IEEE Trans. IT* **15**(1), 122–127 (1969)
12. Moree, P.: Artin's primitive root conjecture—a survey. *arXiv Math arXiv:0412262* (2012)

13. Niederreiter, H., Vielhaber, M.: Simultaneous shifted continued fraction expansions in quadratic time. *AAECC* **9**(2), 125–138 (1998)
14. Perron, O.: *Die Lehre von den Kettenbrüchen*, Bd. I. Teubner, Stuttgart 1954/1977
15. Rueppel, R.: *Analysis and Design of Stream Ciphers*. Springer, Berlin (1986)
16. Tian, T., Qi, W.F.: Expected values for the rational complexity of finite binary sequences. *Des. Codes Crypt.* **55**(1), 65–79 (2010)
17. Vallée, B.: Digits and continuants in euclidean algorithms Ergodic versus tauberian theorems. *J. Théor. Nombres Bordeaux* **12**(2), 531–570 (2000)
18. Vepstas, L.: The Minkowski Question Mark, $PSL(2, \mathbb{Z})$ and the Modular Group <http://ns.linas.org/math/chap-minkowski.pdf> (2020)
19. Vielhaber, M.: Continued fraction expansion as isometry - the law of the iterated logarithm for linear, jump, and 2-adic complexity. *IEEE Trans. IT* **53**(11), 4383–4391 (2007). Preprint: <https://arxiv.org/pdf/cs/0511089.pdf>
20. Vielhaber, M.: A unified view on sequence complexity measures as isometries. In: *SETA 2004 LNCS*, vol. 3486, pp. 143–153 (2005)
21. Vielhaber, M.: Reduce-by-feedback: timing resistant and DPA-aware modular multiplication, plus: how to break RSA by DPA. In: *CHES 2012*. Springer (2012)
22. Vielhaber, M.: V Tree—continued fraction expansion, Stern-Brocot tree. Minkowski's $\psi(x)$ Function In Binary: Exponentially Faster. [arXiv:2008.08020](https://arxiv.org/abs/2008.08020) (2020)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.