Feedback in $\mathbb{Q}$ Shift Registers F$\mathbb{Q}$SR: Pseudo-Ultrametric Continued Fractions in $\mathbb{R}$

Michael Vielhaber
HS Bremerhaven
FB 2
Bremerhaven, Germany
vielhaber@gmail.com

Mónica del P. Canales
MATEMATICVM
@iluminacionmatematica
Valdivia, Chile
monicadelpilar@gmail.com

Sergio Jara C.
Universidad Austral de Chile
Facultad de Ingeniería
Valdivia, Chile
sergio.jara@uach.cl

Abstract

The approximation of binary sequences by ultimately periodic ones can be interpreted in the setting of rational formal power series, *i.e.* from $\mathbb{F}_2(x) \subset \mathbb{F}_2((x^{-1}))$, or rational dyadics, *i.e.* from $\mathbb{Q} \subset \mathbb{Z}_2$, or just rational reals in binary, *i.e.* from $\mathbb{Q} \subset \mathbb{R}$.

The first leads to linear complexity L, LFSRs, and the Berlekamp-Massey algorithm. The second leads to dyadic complexity A, FCSRs, and the Klapper-Goresky Rational Approximation algorithm. The third uses continued fractions over the reals and F$\mathbb{Q}$SRs – introduced here –, and leads to rational complexity R with isometry Q and could be called the Lagrange-Euler approach. We implement the real continued fraction expansion in an "ultrametric", bit-by-bit way, using only add, subtract and shift operations, thus avoiding costly divisions (inversions) and multiplications.

The three complexity measures differ pairwise. While all three ultimately recognize exactly the sequences of the form $uv^\omega \in A^\omega, u \in A^*, v \in A^+$, they do so in a different order and thus attach different complexity profiles to sequences from A^ω.

Keywords: Linear complexity, 2-adic complexity, rational complexity.

1 Introduction

The main result of this paper is a Continued Fraction Expansion (CFE) Algorithm over the reals, which adapts to the ultrametric setting of A^ω. We encode the partial denominators (PD) of the CFE by two codes, C_I and C_{II}, which closely match the behaviour of the Gauß-Kuz'min measure, Section 2. The codes C_I, C_{II} deliver a monotonic selfmap C on A^ω, from sequences to encodings of the CFE's PDs. The Binary CFE, Sections 3, 4 and 8, is obtained bit-by-bit, without costly multiplications or divisions/inversions in $\mathbb{R}$.

In Section 5, we determine the shortest and longest encoding, compared to gain in precision. Having the machinery in place, we can now define the Rational Complexity $R(s, n)$ as the length of the required Feedback in $\mathbb{Q}$ Shift Register (F$\mathbb{Q}$SR) in Section 6 and give some of its characteristics. We give an example, $\pi - 3$, in Section 7.

In Section 9, we compare rational to linear (Berlekamp-Massey Algorithm) [1], [8], [3], [13] and 2-adic [4], [5] complexity. Open Problems are given in Section 10.

2 Gauß-Kuz'min Measure

Gauß, Kuz'min, and Lévy have obtained the following properties of continued fractions in $\mathbb{R}$ and their PDs b_i:

Theorem 1. (*Gauß-Kuz'min-Lévy* [6] [7])
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 average gain in precision, per partial denominator in bits, is* $\dfrac{\pi^2}{6\ln(2)^2} = 3.42371\ldots$, *with* $2^{3.42371/2} = 3.27582\ldots$ *being Lévy's constant.*

3 Binary CFE I: Overview, Codes C_I, C_{II}, Monotonicity

The usual way to compute the CFE of a number $s := \sum_{k\in\mathbb{N}} s_k 2^{-k} \in [0,1) \subset \mathbb{R}$ is:

$$s_0 := s, b_0 := 0, \forall i \in \mathbb{N}\colon\ s_i := \frac{1}{\{s_{i-1}\}} = \frac{1}{s_{i-1} - b_{i-1}}, b_i := \lfloor s_i \rfloor \text{ yields } s = 0 + \frac{1|}{|b_1} + \frac{1|}{|b_2} + \cdots,$$

where the b_i are the partial denominators. The convergents A_1/B_1 to s are obtained via Perron's schema. We use $s = \pi - 3$ as example (since we start with $0 \le s < 1$, no b_0 is used), Table 1:

i	-1	0	1	2	3	4	$\ldots$
b_i			7	15	1	292	$\ldots$
A_i	1	0	1	15	16	4786	$\ldots$
B_i	0	1	7	106	113	33102	$\ldots$

Table 1: Schema for PDs b_i and convergents A_i/B_i.

We have initial values $A_{-1} = B_0 = 1$, $B_{-1} = A_0 = 0$ and then $A_i := b_i \cdot A_{i-1} + A_{i-2}$, $B_i := b_i \cdot B_{i-1} + B_{i-2}$ with $|s - A_i/B_i| < B_i^{-2}$ for $s \notin \mathbb{Q}$, [11, Satz 2.10].

Notation:
$\mathbb{N} = \{1,2,3,\ldots\}, \mathbb{N}_0 = \{0,1,2,3,\ldots\}$
$\mathbb{D} := \{a/2^k\colon a \in \mathbb{Z} \text{ odd}, k \in \mathbb{N}_0\}$, dyadic fractions
For $X \subset \mathbb{R}$, $X_1 := X \cap [0,1) : \mathbb{D}_1, \mathbb{Q}_1, \mathbb{R}_1$
$A = \{0,1\}$ is the binary alphabet, 0/1-inversion: $\overline{0} = 1$, $\overline{1} = 0$, $\overline{10011} = 01100$
$A^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$ and A^ω are the finite, resp. infinite words over A
For $v = v_1 v_2 \ldots v_{|v|} \in A^*$: $(v)_2 = \sum_{k=0}^{|v|-1} v_{|v|-k} 2^k \in \mathbb{N}_0$ in binary, $(\varepsilon)_2 = 0$

In the case of linear complexity, the Berlekamp-Massey algorithm (BMA) [1][8] with the adaptations of Dornstetter [3] and Vielhaber [13] takes a sequence s of coefficients of a formal power series

$$\mathbb{F}_2((x^{-1})) \ni G(s) = \sum_{k=1}^{\infty} s_k x^{-k} = \frac{1 \ \ |}{|p_1(x)} + \frac{1 \ \ |}{|p_2(x)} + \frac{1 \ \ |}{|p_3(x)} + \cdots,$$

whose continued fraction expansion (CFE) with partial denominators (PD) $p_i(x) \in \mathbb{F}_2[x]$ determines the linear complexity profile of s. The BMA computes an isometry $\mathbf{K}$ on $\mathbb{F}_2^{\omega}$ such that

$$\mathbf{K}(s) = d = \mathrm{C}_{\mathbb{F}_2[\mathrm{x}]}(p_1)|\mathrm{C}_{\mathbb{F}_2[\mathrm{x}]}(p_2)|\mathrm{C}_{\mathbb{F}_2[\mathrm{x}]}(p_3)|\ldots \in \mathbb{F}_2^{\omega},$$

with $\mathrm{C}_{\mathbb{F}_2[\mathrm{x}]}\colon \mathbb{F}_2[x]\backslash\mathbb{F}_2 \to \mathbb{F}_2^{2d}, p(x) = \sum_{k=0}^{d} a_k x^k \mapsto \mathrm{C}_{\mathbb{F}_2[\mathrm{x}]}(p(x)) = 0^{d-1}1a_{d-1}\ldots a_1 a_0$, yields the discrepancy sequence d, which *at the same time* is an encoding of the PDs of $G(s)$. For fast implementations see [2] [9].

Our approach is inspired by this ultrametric, bitwise model and will avoid both the costly inversion $s_i := 1/\{s_{i-1}\}$, and the multiplications $b_i \cdot A_{i-1}, b_i \cdot B_{i-1}$. Instead, we do a binary search for the CFE by encoding the PDs of a trial number $r = [b_1, \ldots, b_{2i}]$ generated by an FQSR (Section 8), $c = C(r), r = C^{-1}(c)$.

Our encoding $C\colon \mathbb{R}_1 \to A^{\omega}\backslash A^*1^{\omega}$ will satisfy three properties:
(GK): C is reasonably close to the Gauß-Kuz'min measure.
(MON): C is monotonically increasing, $r < r' \Longleftrightarrow C(r) < C(r')$.
(INC): Incremental, from $r = \dfrac{A_i}{B_i} = C^{-1}(v|10^{\omega})$, we get $C^{-1}(v|a|10^{\omega}) = \dfrac{A_i \pm (A_{i-1} << \delta)}{B_i \pm (B_{i-1} << \delta)}$
for both $a \in \{0,1\}$ and some $\delta \in \mathbb{N}_0$. Hence, all adjustments of A_i, B_i are made by adding a shifted copy of A_{i-1}, B_{i-1}, without multiplication.

Definition 2. Encodings $\mathrm{C_I}$, $\mathrm{C_{II}}$, $C(r)$ (see Table 2)
For $b \in \mathbb{N}$, let $n := \lfloor \log_2(b) \rfloor$, $b = 2^n + \sum_{k=0}^{n-1} a_k 2^k$.
(i) For PDs with odd index, let

$$\mathrm{C_I}\colon \mathbb{N} \to A^*, \ b \mapsto \mathrm{C_I}(b) = 0^n 1\overline{a}_{n-1}\ldots\overline{a}_1\overline{a}_0,$$

in particular, $1 \mapsto 1; 2 \mapsto 011; 3 \mapsto 010$. We also set $\mathrm{C_I}\colon \aleph_0 \mapsto 0^{\infty}$ to finish a finite PD sequence, yielding an infinite code $C(r) \in A^{\omega}$, without effect on the value of r.

(ii) For PDs with even index, we 0/1-invert the codewords to obtain code $\mathrm{C_{II}}$:

$$\mathrm{C_{II}}\colon \mathbb{N} \to A^*, \ b \mapsto \mathrm{C_{II}}(b) = 1^n 0 a_{n-1}\ldots a_1 a_0,$$

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

(iii) We concatenate encodings (rational finite CFE and irrational infinite CFE) as
$C : \mathbb{Q}_1 \to A^*0^{\omega}, r = [b_1, b_2, \ldots, b_{2l}, \aleph_0] \mapsto C(r) := \mathrm{C_I}(b_1)|\mathrm{C_{II}}(b_2)|\ldots|\mathrm{C_{II}}(b_{2l})|\mathrm{C_I}(\aleph_0)$,
$C : \mathbb{R}_1\backslash\mathbb{Q}_1 \to A^{\omega}\backslash(A^*0^{\omega} \cup A^*1^{\omega}), r = [b_1, b_2, b_3, \ldots] \mapsto C(r) := \mathrm{C_I}(b_1)|\mathrm{C_{II}}(b_2)|\mathrm{C_I}(b_3)|\ldots$.

(GK): Columns $l_{\mathrm{I,II}}$ with $l_{\mathrm{I,II}}(b) = 1 + 2\lfloor \log_2(b) \rfloor$ and l_{GK} with $l_{GK}(b) = -\log_2(\mu_{GK}(b))$ of Table 2 give the coding length for our code and an "ideal code" along the Gauß-Kuz'min measure (with non-integral word lengths), respectively.

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	—	—	—

Table 2: Codes C_I and C_{II} for partial denominators.

Lemma 3. *Information Content of prefixes from $C(s)$ vs. $s, d = Q(s)$*

Comparing the length distributions of the two codes, for numbers satisfying μ_{GK}, for $n \to \infty$, the encoding of $[b_1, \ldots, b_{2i}] = A_{2i}/B_{2i} = r$ matching n bits of s, i.e. $r_k = s_k, \forall 1 \leq k \leq n$ has a length $\sum_{k=1}^{2i} l_{I,II}(b_k) \approx 1.024 \cdot n$. That is, the encoding $C(s)$ grows faster than the considered prefix length of the sequences s and $d = Q(s)$ by a factor of (only) 1.024.

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

From continued fraction theory, we know the impact on r of changing a PD:

Proposition 4. *Let two real numbers $r' = [b_1, \ldots, b_{i-1}, b'_i, \ldots]$, $r'' = [b_1, \ldots, b_{i-1}, b''_i, \ldots]$ with $b'_i < b''_i$ be given. Then $r' < r''$ if and only if i is even.*

Proof. See Satz 2.8 in Perron [11]. □

Theorem 5. (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 lexicographic order.*

Proof. (hint) Follows with Proposition 4. As mentioned (see [11, Satz 2.8]), for odd index $2i+1$ ($b_j, j < 2i+1$, fixed), $b_{2i+1} > b'_{2i+1}$ implies $r = [b_1, \ldots, b_{2i+1}] < r' = [b_1, \ldots, b_{2i}, b'_{2i+1}]$. Also, we then have $\mathrm{C_I}(b_{2i+1}) < \mathrm{C_I}(b'_{2i+1})$ and $C(r) < C(r')$ lexicographically. For even index $2i+2$, we have $b_{2i+2} < b'_{2i+2} \Leftrightarrow r < r' \Leftrightarrow \mathrm{C_{II}}(b_{2i+2}) < \mathrm{C_{II}}(b'_{2i+2}) \Leftrightarrow C(r) < C(r')$. □

We have to deal with two ambiguities. One is $v01^\omega \equiv v10^\omega, \forall v \in A^*$ both as binary representations (of the same real number) and as encoding of PDs, with $v01^\omega =$

```
Binary CFE with FQSR

v := ε   // code[b_i]
s := ε   // input sequence
d := ε   // discrepancies
         // d = Q(s)
r := 0^ω // approximation
         // r = C^{-1}(v)
k := 0   // index in s, d, r
FOREVER
    k := k + 1
    read_next_bit s_k
    IF s_k = r_k
       // Case ≡
       d_k := 0
    ELSE
       d_k := 1
       WHILE (s_k ≠ r_k)
          IF s_k < r_k
             // Case ⊕
             v := v|0
          ELSE // s_k > r_k
             // Case ⊖
             v := v|1
          ENDIF
          (b_i) := decode(v10^∞)
          (A, B) := perron((b_i))
          r := A/B // by FQSR
       ENDWHILE
    ENDIF
ENDFOR
```

$v.10^\infty$	PDs	A/B	r
.1|0|	1|1|	1/2	$.10^\omega$
0.10|0|	3|1|	1/4	.01000(0)
|1.00|	1|2|	2/3	.(10)
00.100|0|	7|1|	1/8	.001(0)
01.1|0|	2|1|	1/3	.(01)
1|0|.1|0|	1|1|1|1|	3/5	.(1001)
1|1.1000|	1|4|	4/5	.(1100)
000.1000|0|	15|1|	1/16	.0001(0)
001.10|0|	5|1|	1/6	.00(10)
010|.100|	3|2|	2/7	.(010)
011|.100|	2|2|	2/5	.(0110)
1|0|0.10|0|	1|1|3|1|	5/9	.(100011)
1|0|1|.100|	1|1|1|2|	5/8	.101(0)
1|10.1|	1|3|	3/4	.11(0)
1|11.10000|	1|8|	8/9	.(111000)
0000.10000|0|	31|1|	1/32	.00001(0)
0001.100|0|	11|1|	1/12	.00(01)
0010.1|0|	6|1|	1/7	.(001)
0011.1|0|	4|1|	1/5	.(0011)
010|0.|1|0|	3|1|1|1|	3/11	.(0100010111)
010|1.1000|	3|4|	4/13	.(010011101100)
011|0|.1|0|	2|1|1|1|	3/8	.011(0)
011|1.1000|	2|4|	4/9	.(011100)
1|0|00.100|0|	1|1|7|1|	9/17	.(10000111)
1|0|01.1|0|	1|1|2|1|	4/7	.(100)
1|0|1|0|.1|0|	1|1|1|1|1|1|	8/13	.(100111011000)
1|0|1|1.1000|	1|1|1|4|	9/14	.1(010)
1|100|.1|0|	1|2|1|1|	5/7	.(101)
1|101|.1|0|	1|3|1|1|	7/9	.(110001)
1|110.10|	1|6|	6/7	.(110)
1|111.0000|	1|16|	16/17	.(11110000)

Table 3: Binary CFE and approximations.

$C(\ldots, b_{2i}, 1, \aleph_0)$ and $v10^\omega = C(\ldots, b_{2i}+1, \aleph_0)$, or $v01^\omega = C(\ldots, b_{2i+1}, \aleph_0)$ and $v10^\omega = C(\ldots, b_{2i+1}-1, 1, \aleph_0)$.

A dyadic fraction $r = A_i/B_i$ from the FQSR is received as $.v10^\omega$ and so matches a sequence s with rational complexity $|v|+1$ (see Section 6). For $s = .v01^\omega$ (with the same value in $\mathbb{R}_1$), we assign the rational complexity $|v|+2$ as a special case.

The other ambiguity $[\ldots, b_i, 1] = [\ldots, b_i + 1]$ can be resolved in 4 ways: Let the last PD be (i) always greater than 1, or (ii) always 1, or have (iii) an even, or (iv) an odd number of PDs. We shall use convention (iii), and thus the encoding C always terminates in 10^ω.

Table 3 (left) shows how to calculate $C(s)$ incrementally, avoiding costly inversions of s by instead comparing r to s and choosing the next encoding $v0$ or $v1$ accordingly.

$c = C(r)$ will eventually settle into the CFE $[b'_1, \ldots, b'_i, \ldots]$ of s, with more and more PDs becoming correct and stable. Each PD b_i runs through the trial values $1, 2, 4, \ldots, 2^k, \ldots$ until $2^{k-1} < b'_i \le 2^k$ and is then adjusted towards b'_i by interval halving, starting with $b_i := 2^{k-1} + 2^{k-2}$ as midpoint of the interval $[2^{k-1}, 2^k]$.

We also compute an isometry Q on A^ω, the discrepancy sequence $d = \mathrm{Q}(s)$. As long as $s_k = r_k$, we keep on reading in s_{k+1} and producing r_{k+1} by the FQSR, Case $\equiv$, with discrepancy $d_k := 0$. For rational, ultimately periodic s, this goes on forever.

Otherwise, either $0 = s_k < r_k = 1$ for some index, and thus r is larger than s (Case $\oplus$), or else $0 = r_k < s_k = 1$, with $r < s$ (Case $\ominus$). We extend the code v to $v|0$ or $v|1$, respectively, with $C^{-1}(v|0|10^\omega) < C^{-1}(v|10^\omega) < C^{-1}(v|1|10^\omega)$, and set $d_k = \mathrm{Q}(s)_k := 1$.

On the right-hand side of Table 3, we show all possible encodings for $|v| \le 4$, *i.e.* the first 5 approximation steps for any sequence s.

4 Binary CFE II: States, Incremental Change of b_i, A_i, B_i

We start with the full cylinder set A^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 $v10^\omega \mapsto r$, which sits at the center 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

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

Remark 6. Referring to [16], the encodings $v10^\omega$ pass through the dyadic fractions from $\mathbb{D}_1$ on the van der Corput tree, while the resulting approximations $A/B \in \mathbb{Q}_1$ are labels of the V_{10} tree. Hence, we just calculate the inverse V question mark function $r := ?_V^{-1}(\iota_{AD}(v))$ (see [16, Defs. 2 and 3, Appx. 4]).

In order to obtain an efficient implementation of this part, we maintain the state of a Finite State Machine + Counter (FSM+C) and the current as well as an auxilliary convergent, A/B and A'/B', respectively. The state set is $\{\mathtt{A}\ldots\mathtt{D}, \overline{\mathtt{A}}\ldots\overline{\mathtt{D}}\}$, the counter appears as index k for states $\mathtt{B}, \mathtt{C}, \overline{\mathtt{B}}, \overline{\mathtt{C}}$.

	IN: $v.10^\infty$	OUT $\oplus : v0.10^\infty$, for $r > s$ OUT $\ominus : v1.10^\infty$, for $r < s$	
A	$.1\|0\| \equiv [1,1]$	$0.10\|0\| \equiv [3,1]$	$\mathtt{B}_1$
		$1\|.100\| \equiv [1,2]$	$\overline{\mathtt{A}}$
$\mathtt{B}_k$	$0^k.10^k\|0\| \equiv [2^{k+1}-1,1]$	$0^k0.10^{k+1}\|0\| \equiv [2^{k+2}-1,1]$	$\mathtt{B}_{k+1}$
		$0^k1.10^{k-1}\|0\| \equiv [2^k+2^{k-1}-1,1]$	$\mathtt{C}_{k-1}, \mathtt{D}$
$\mathtt{C}_k$ $k\geq 2$	$0^{k+m}1*^m.10^{k-1}\|0\| \equiv [N,1]$	$0^{k+m}1*^m 0.10^{k-2} \equiv [N+2^{k-2},1]$	$\mathtt{C}_{k-1}, \mathtt{D}$
		$0^{k+m}1*^m 1.10^{k-2} \equiv [N-2^{k-2},1]$	$\mathtt{C}_{k-1}, \mathtt{D}$
D $= \mathtt{C}_1$	$0^k1*^{k-1}.1\|0\| \equiv [N,1]$	$0^k1*^{k-1}0\|.100\| \equiv [N+1,2]$	$\overline{\mathtt{A}}$
		$0^k1*^{k-1}1\|.100\| \equiv [N,2]$	$\overline{\mathtt{A}}$
$\overline{\mathtt{A}}$	$\ldots\|.100\| \equiv [\ldots,2]$	$\ldots\|0\|.1\|0\| \equiv [\ldots,1,1,1]$	A
		$\ldots\|1.1000\| \equiv [\ldots,4]$	$\overline{\mathtt{B}}_1$
$\overline{\mathtt{B}}_k$	$\ldots\|1^k.10^{k+2}\| \equiv [\ldots,2^{k+1}]$	$\ldots\|1^k0.10^{k-1} \equiv [\ldots,2^k+2^{k-1}]$	$\overline{\mathtt{C}}_k, \overline{\mathtt{D}}$
		$\ldots\|1^k1.10^{k+3} \equiv [\ldots,2^{k+2}]$	$\overline{\mathtt{B}}_{k+1}$
$\overline{\mathtt{C}}_k$ $k\geq 2$	$\ldots\|1^{k+m}0*^m.10^{k-1}\| \equiv [\ldots,N]$	$\ldots\|1^{k+m}0*^m 0.10^{k-2}\| \equiv [\ldots,N-2^{k-2}]$	$\overline{\mathtt{C}}_{k-1}, \overline{\mathtt{D}}$
		$\ldots\|1^{k+m}0*^m 1.10^{k-2}\| \equiv [\ldots,N+2^{k-2}]$	$\overline{\mathtt{C}}_{k-1}, \overline{\mathtt{D}}$
$\overline{\mathtt{D}}$ $= \overline{\mathtt{C}}_1$	$\ldots\|1^k0*^{k-1}.1\| \equiv [\ldots,N]$	$\ldots\|1^k0*^{k-1}0\|.1\|0\| \equiv [\ldots,N-1,1,1]$	A
		$\ldots\|1^k0*^{k-1}1\|.1\|0\| \equiv [\ldots,N,1,1]$	A

Table 4: Encoding of states.

State	$A =$	$a = 0$		$a = 1$	
	$A_{\ldots}$	$A'^+ :=$	$A^+ := \ldots$	$A'^+ :=$	$A^+ := \ldots$
A	$2i+1$	A'	$A + (A' << 1)$	$A - A'$	$(A<<1) - A'$
$\mathtt{B}_k$	$2i+1$	A'	$A + (A' << (k+1))$	A'	$A - (A' << (k-1))$
$\mathtt{C}_k$	$2i+1$	A'	$A + (A' << (k-2))$	A'	$A - (A' << (k-2))$
D	$2i+1$	A	$(A << 1) + A'$	$A - A'$	$(A << 1) - A'$
$\overline{\mathtt{A}}$	$2i+2$	$A - A'$	$(A << 1) - A'$	A'	$A + (A' << 1)$
$\overline{\mathtt{B}}_k$	$2i+2$	A'	$A - (A' << (k-1))$	A'	$A + (A' << (k+1))$
$\overline{\mathtt{C}}_k$	$2i+2$	A'	$A - (A' << (k-2))$	A'	$A + (A' << (k-2))$
$\overline{\mathtt{D}}$	$2i+2$	$A - A'$	$(A << 1) - A'$	A	$(A << 1) + A'$

Table 5: Update of convergents.

The states A – D handle PDs with odd index, while states $\overline{\mathtt{A}}$ – $\overline{\mathtt{D}}$ for PDs with even index operate exactly 0/1-inverse to A – D. Since we always maintain an even number of PDs, for an odd current index the PD value is distributed as $[\ldots, b_{2l-1} - 1, 1]$. This is the only difference between the two groups of states, apart from the 0/1 inversion. The update of the state and the two convergents are shown in Tables 4 and 5 (for B, B' analogously). The first $2i$ PDs are fixed and omitted. Observe that no multiplication (as in $A_i := b_i \cdot A_{i-1} + A_{i-2}, B_i := b_i \cdot B_{i-1} + B_{i-2}$) is necessary.

(INC): Everything is done bitwise, incrementally, and only additions and shifts are used.

5 Maximum Coding Gain/Loss

Lemma 3 shows that for numbers satisfying the Gauß-Kuz'min measure, the code grows by a factor 1.024 faster than the representation. For numbers not satisfying the Gauß-Kuz'min measure, the code may grow faster or slower. We compare the coding length l for $\Phi_b = [b, b, b, b, \dots]$ with its gain in precision. In this case, the convergents' denominators (and numerators) asymptotically increase by a factor $\varphi_b = (b + \sqrt{b^2+4})/2$ $(= b + \Phi_b)$, from $\begin{array}{c|c|c} & & b_i = b \\ \hline 1 & \varphi_b & b \cdot \varphi_b + 1 \end{array}$ (Perron's schema) with $b \cdot \varphi_b + 1 \stackrel{!}{=} \varphi_b^2 \iff \varphi_b^2 - b\varphi_b - 1 = 0$.

Since $|r - A_i/B_i|$ decreases like $\Theta(B_i^{-2}) = \Theta(\varphi_b^{-2i})$ (for $r \notin \mathbb{Q}$), see [11, Satz 2.10], each additional PD b asymptotically yields $L := \log_2(\varphi_b^2)$ more bits with $r_k = s_k$. As b is encoded by $l = l_{\mathrm{I,II}}(b) = 1 + 2\lfloor \log_2(b) \rfloor$ bits, we obtain the coding gain/loss by

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

The values γ_b decrease monotonically with b from $b = 2^k$ to $b = 2^{k+1} - 1$ with fixed $l = 2k + 1$, then jump to a local maximum at 2^{k+1} etc. From Table 6, we can infer that $\gamma_1 = 0.72$ is the minimum, $\gamma_4 = 1.20$ the maximum value. Hence, φ_1 has the shortest, φ_4 the longest code, compared with attained precision.

b	φ_b	L	l	γ_b	γ_b [±%]	$1/\gamma_b$ [±%]	b	φ_b	L	l	γ_b	γ_b [±%]	$1/\gamma_b$ [±%]
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
A_i	1	0	1	2	3	5	8	13	21	34	55
B_i	0	1	2	3	5	8	13	21	34	55	89
A_i	1	0	1	4	17	72	305	1292	…		
B_i	0	1	4	17	72	305	1292	5473	…		

$$\varphi_1\colon \frac{l}{L} = \frac{(9 \times 1)}{\log_2(89^2)} = 0.69 \approx \gamma_1$$

$$\varphi_4\colon \frac{l}{L} = \frac{(6 \times 5)}{\log_2(5473^2)} = 1.208 \approx \gamma_4$$

Table 6: Gain γ_b in precision for fixed partial denominator. CFE for φ_1, φ_4.

6 Rational Complexity of a Binary Sequence

In analogy to linear and 2-adic complexity, we define the rational complexity of a sequence:

Definition 7. Rational Complexity $R(s, n)$

(*i*) Let $R(0^\omega, n) = R(0^k, n) := 0, \forall n \le k \in \mathbb{N}_0$. Otherwise, the rational complexity $R(n) := R(s, n) \in \mathbb{N}_0$ of $s \in A^* \cup (A^\omega \backslash A^* 1^\omega)$ is $R(n) := \lceil \log_2(B_i) \rceil$, where $A_i/B_i = r$ is

the first convergent of the binary CFE that matches $r_k = s_k, 1 \le k \le n$. Thus $R(n)$ is the shortest length of an FQSR producing the n-bit prefix of s. Set $R(1^\omega, n) = 1, \forall n$ and for $s = v01^\omega$, set $R(s, n) := R(v0, n), n \le |v| + 1$ and $R(s, n) := |v| + 2$ for $n \ge |v| + 2$.

(*ii*) For a sequence $s \in A^\omega$, we define its rational complexity profile as $(R(n)_{n\in\mathbb{N}}) \in \mathbb{N}_0^\omega$.

The distribution of rational complexities for words of size up to 12, as well as expectation and variance up to length 24 are given in Table 7, $E = E(R(n))$, $\mathrm{Var} = \mathrm{Var}(R(n))$.

n	E	Var	q:	0	1	2	3	4	5	6	7	8	9	10	11	12	13
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	1	
12	6.892	1.2556		1	2	5	18	62	252	942	1835	731	186	48	12	2	1

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

Table 7: Rational complexity, $N_R(n, q)$, $\mathrm{E}(R(n))$, $\mathrm{Var}(R(n))$.

Theorem 8. *Rational Complexity: Counts*

The number $N_R(q) = \lim_{n\to\infty} N_R(n, q)$ *of sequences with finite rational complexity* $R(s, n) = q$ *for* $n \to \infty$ *is* $N_R(0) = 1$ $(s = 0^\omega)$, $N_R(1) = 2$ $(s = 10^\omega, 1^\omega)$, *and for* $q \ge 2$

$$N_R(q) := 2^{q-2} + \sum_{B=2^{q-1}+1}^{2^q} \phi(B),$$

where $\phi(B)$ *is Euler's totient function.*

Proof. The sum is the number of fractions A/B with $1 \le A < B, 2^{q-1} < B \le 2^q$ and A, B coprime. The additional 2^{q-2} sequences correspond to the special cases $*^q01^\omega$ for dyadic fractions. □

Conjecture 9. Rational Complexity: Average and Variance

Distinguishing even and odd lengths n, we have two values each for $\mathrm{E}(R(n)) \approx n/2 + 1 - 0.06$ (where the nature of 0.06 is yet unclear), and $\mathrm{Var}(R(n)) \approx 1.25 \pm o(n)$.

7 Example π

The well-known CFE of π = 0x3.243F6A8885A308D... is $[3; 7, 15, 1, 292, 1, \dots]$. Our binary algorithm proceeds on $\pi - 3$ as given in Table 8. We omit the final code 0^ω and PD $\aleph_0$, present in all lines. $S \in \{\mathtt{A}, \dots, \overline{\mathtt{D}}\}$ is the state. k gives the first place with $s_k \neq r_k$, and $\oplus$ indicates $r > s$, while for $\ominus$, $r < s$.

read	S	$c.10^\omega$	$[(b_i)]$	p/q	r	k	$\oplus/\ominus$
	$\mathtt{A}$	.1|0|	$[1, 1]$	1/2	8000	1	$\oplus$
s_1	$\mathtt{B}_1$	0.10|0|	$[3, 1]$	1/4	4000	2	$\oplus$
s_2	$\mathtt{B}_2$	00.100|0|	$[7, 1]$	1/8	2000	6	$\ominus$
$s_3 \dots s_6$	$\mathtt{C}_2$	001.10|0|	$[5, 1]$	1/6	2AAA	5	$\oplus$
	$\mathtt{D}$	0010.1|0|	$[6, 1]$	1/7	249249	9	$\oplus$
$s_7 \dots s_9$	$\overline{\mathtt{A}}$	00100|.100|	$[7, 2]$	2/15	2222	6	$\ominus$
	$\overline{\mathtt{B}}_1$	00100|1.1000|	$[7, 4]$	4/29	234	6	$\ominus$
	$\overline{\mathtt{B}}_2$	00100|11.10000|	$[7, 8]$	8/57	23E	6	$\ominus$
	$\overline{\mathtt{B}}_3$	00100|111.100000|	$[7, 16]$	16/113	243F6F	22	$\oplus$
$s_{10} \dots s_{22}$	$\overline{\mathtt{C}}_3$	00100|1110.100|	$[7, 12]$	12/85	242424	12	$\ominus$
	$\overline{\mathtt{C}}_2$	00100|11101.10|	$[7, 14]$	14/99	2433B	13	$\ominus$
	$\overline{\mathtt{D}}$	00100|111011.1|	$[7, 15]$	15/106	2439F	14	$\ominus$
	$\mathtt{A}$	00100|1110111.|1|0|	$[7, 15, 1, 1]$	31/219	243CC	15	$\ominus$
	$\overline{\mathtt{A}}, \overline{\mathtt{B}}_{1..6}$	$\vdots$	$[7, 15, 1, 2^k]$		$\vdots$		$\ominus$
	$\overline{\mathtt{B}}_7$	00..|..11|1|$1^7.10^9$|	$[7, 15, 1, 256]$		243F69E52	23	$\ominus$
s_{23}	$\overline{\mathtt{B}}_8$	00..|..11|1|$1^8.10^{10}$|	$[7, 15, 1, 512]$		243F6C728	22	$\oplus$
	$\overline{\mathtt{C}}_8$	00..|..11|1|$1^8$0.10000000|	$[7, 15, 1, 384]$		243F6B987	24	$\oplus$
s_{24}	$\overline{\mathtt{C}}_7$	00..|..11|1|$1^8$00.1000000|	$[7, 15, 1, 320]$		243F6AEA3	26	$\oplus$
s_{25}, s_{26}	$\overline{\mathtt{C}}_6$	00..|..11|1|$1^8$000.100000|	$[7, 15, 1, 288]$		243F6A762	25	$\ominus$
	$\overline{\mathtt{C}}_5$	00..|..11|1|$1^8$0001.10000|	$[7, 15, 1, 304]$		243F6AB33	27	$\oplus$
s_{27}	$\overline{\mathtt{C}}_4$	00..|..11|1|$1^8$00010.1000|	$[7, 15, 1, 296]$		243F6A958	28	$\oplus$
s_{28}	$\overline{\mathtt{C}}_3$	00..|..11|1|$1^8$000100.100|	$[7, 15, 1, 292]$		243F6A860	29	$\ominus$
s_{29}	$\overline{\mathtt{C}}_2$	00..|..11|1|$1^8$0001001.10|	$[7, 15, 1, 294]$		243F6A8DD	30	$\oplus$
s_{30}	$\overline{\mathtt{D}}$	00..|..11|1|$1^8$00010010.1|	$[7, 15, 1, 293]$		243F6A89F2	32	$\oplus$
s_{31}, s_{32}	$\mathtt{A}$	00..|..|1|..100|.1|0|	$[7, 15, 1, 292, 1, 1]$		243F6A87FF	29	$\ominus$
	$\overline{\mathtt{A}}$	00..|..|1|..100|1|.100|	$[7, 15, 1, 292, 1, 2]$		243F6A88A5	35	$\oplus$
$s_{33} \dots s_{35}$	$\mathtt{A}$	00..|..|1|..100|1|0|.1|0|	$[7, 15, 1, 292, 1, 1, 1, 1]$		243F6A8863	36	$\oplus$

Table 8: Approximating π

Given that the sequence $\pi - 3 =$ .243F6A88... fails to meet the suggested approximations in places $s_1, s_2, s_6, s_9, s_{22-24}, s_{26-30}, s_{32}, s_{35}$ etc., we obtain the discrepancy sequence Q(0010.0110.0011.1111.0110.1010.1000.1000) = 1100.0100.1000.0000.0000.0111.0111.1101 = 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 (or 355/113 for π).

Also, $C(s)$ = 00100.1110111.1.11111111000100100.1... (compare with the $\oplus/\ominus$ sequence).

8 FQSRs: Hardware Implementation

We have to obtain the rational sequence $r = A/B$ for comparison with s.
Paper-and-pencil long division of A by B (here $A < B$ is assumed) proceeds as follows:

$X := A$
FOREVER
  $X := 2 \cdot X$
  IF $X \geq B$ THEN Output 1; $X := X - B$ ELSE Output 0 ENDIF
ENDFOR

We simulate division in $\mathbb{Q}$ by Reduce-By-Feedback (see [15]) in an FQSR, a standard long accumulator X for values in $\mathbb{N}_0$ with a shift-and-add unit.

Let $n := \lceil \log_2(B) \rceil, N := 2^n$, hence $N/2 < B \leq N$. We use an $(n+2)$-bit register for X, initially filled with A. Setting $K := (N \mod B) = N - B$, we may replace $X := X - B$ by $X := X - N + K$. We furthermore replace the decision $X \geq B$? (long compare) by the one-bit-check $X \geq N? \equiv (X_n = 1)?$ and, if necessary, remove the n-th bit 1, an operation worth $X := X - N$, while adding the constant K.

The two leftmost FQSR bits, of content $\mu := 0, 1, 2$, or 3, indicate the multiple of N to be removed and replaced by $\mu \cdot K$. We now show that $\mu \leq 2$ is always guaranteed.

Theorem 10. *Overflow-freeness of the range $\mu \in \{0, 1, 2\}$*
Let $0 \leq \mu \leq 2$. Then $0 \leq 2 \cdot X + \mu \cdot K < 3 \cdot N$ and thus again $0 \leq \lfloor (2X + \mu K)/N \rfloor \leq 2$.

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

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

$ctr := 0; k := 0$
FOR $l = 1, 2, \ldots$
  IF $\mu_l = 0$ THEN Output $(r_k = 0, r_{k+1} = \ldots r_{k+ctr} = 1); k := k + 1 + ctr; ctr := 0$ ENDIF
  IF $\mu_l = 1$ THEN $ctr := ctr + 1$ ENDIF
  IF $\mu_l = 2$ THEN Output $(r_k = 1, r_{k+1} = \ldots r_{k+ctr} = 0); k := k + 1 + ctr; ctr := 0$ ENDIF
ENDFOR

Essentially, a run of ones with leading zero is saved for later and output when $\mu = 0$. With $\mu = 2$, 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 $A < B$, we do never start with $\mu = 1^k 2 \ldots$).

More on the implementation of FQSRs in hardware can be found in [15, 12], 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.

9 Comparison of Q with A and L

Rational complexity R with isometry Q, linear complexity L [1], [8], [3], [13], and 2-adic complexity A [4],[5], all assign a finite complexity to ultimately periodic sequences. The order of appearance of the rational strings is, however, quite different.

Table 9 shows, how many of the 2^m simplest sequences according to one measure are contained in the set of 2^n simplest sequences of another measure, for $m = 4, \ldots, 28$ by 4 and both $n = m$ and $n = 28$, for all 3 or 6 (un-/ordered) pairs (Y, Z) from $\{\mathrm{Q}, \mathrm{A}, \mathrm{L}\}$. Figures given are $\log_2 |\,\mathrm{Y}[1 \ldots 2^m] \cap \mathrm{Z}[1 \ldots 2^n]|$.

Apparently, $\log_2 |\,\mathrm{Y}[1 \ldots 2^m] \cap \mathrm{Z}[1 \ldots 2^m]| = O(0.6m+2)$ asymptotically. That is, only about $4N^{0.6}$ of the first N sequences will match for any two isometries.

	$m = n = 4, \ldots, 28$				$m = 4, \ldots, 28, n = 28$					
m	$\lvert \mathrm{L} \cap \mathrm{Q} \rvert$	$\lvert \mathrm{L} \cap \mathrm{A} \rvert$	$\lvert \mathrm{Q} \cap \mathrm{A} \rvert$	m	$\lvert \mathrm{L} \cap \mathrm{Q} \rvert$	$\lvert \mathrm{Q} \cap \mathrm{L} \rvert$	$\lvert \mathrm{L} \cap \mathrm{A} \rvert$	$\lvert \mathrm{A} \cap \mathrm{L} \rvert$	$\lvert \mathrm{Q} \cap \mathrm{A} \rvert$	$\lvert \mathrm{A} \cap \mathrm{Q} \rvert$
4	3.32	3.32	3.32	4	4.00	4.00	4.00	4.00	4.00	4.00
8	6.43	6.21	6.67	8	7.59	7.92	7.52	7.98	7.89	7.98
12	9.24	9.15	9.62	12	10.55	11.00	10.49	11.32	10.98	11.31
16	11.77	11.85	12.24	16	13.11	13.40	13.09	13.85	13.48	13.92
20	14.19	14.35	14.71	20	15.35	15.47	15.37	15.94	15.66	16.07
24	16.54	16.74	17.06	24	17.31	17.37	17.40	17.76	17.67	18.00
28	18.83	19.09	19.39	28	18.83	18.83	19.09	19.09	19.39	19.39

Table 9: Intersection size, given as $\log_2 |Y \cap Z|$.

A further comparison of the structure of the 3 isometries $\mathrm{Y} \in \{\mathrm{Q}, \mathrm{A}, \mathrm{L}\}$ can be made by determining their tree complexity, see [10] [14].

Let an infinite regular binary tree be indexed by $v \in A^*$ with root $v = \varepsilon$. Each node v has its left and right child nodes indexed as $v0$ and $v1$, respectively, and is given a label $\widehat{\mathrm{Y}}(v) = \mathrm{Y}(v0^\omega)_{|v|+1} \in A$. The tree complexity (of the labeling) then is given in Table 10.

$\widehat{Y}$	h: 1	2	3	4	5	6
$\widehat{\mathrm{L}}$	2	8	48	480	2816	21760
$\widehat{\mathrm{Q}}$	2	8	118	12244	2195K	45M
$\widehat{\mathrm{A}}$	2	8	128	10506	1931K	91M

Table 10: Tree complexities of induced isometries.

Apparently, L is by far the least complex (maybe: best understood) of the three isometries, while Q and A are of comparable complexity, with Q slightly simpler judging from the entry for height $h = 6$. This coincides with $|Q \cap A|$ being larger, or Q and A being nearer to each other than to K, in Table 9.

10 Open Problems

1. Whenever the current approximation is very good (like after s_9 for π, with $d_{10} = \cdots = d_{21} = 0$), how can we use this fact to accelerate the algorithm?
2. How, if at all, can the binary procedure be adapted to sequences over $\mathbb{F}_p$, $p > 2$?
3. After a change of A_i, B_i, an LFSR can be restarted using the last n output bits. Is there an equivalent procedure for FQSRs?
4. Give closed formulae for the rational complexity values $N_R(n, q)$, $E(R(n))$ and $\mathrm{Var}(R(n))$, for even and odd n. Describe the number 0.06 in $E(R(n)) = n/2 + 1 - 0.06$ (Section 6).
5. What can be said about T-periodic sequences? Does the complexity depend on ord(2) in $\mathbb{Z}/T\mathbb{Z}$?
6. How does the rational complexity behave for sequences with perfect autocorrelation, in particular compared with linear complexity and 2-adic complexity?

Conclusion

We have introduced Rational Complexity as a third way to assess the non/randomness of binary sequences, besides linear and 2-adic complexity. The algorithm is a binary, pseudo-ultrametric version of the standard continued fraction expansion in $\mathbb{R}$, making use of FQSRs, Feedback in $\mathbb{Q}$ Shift Registers. The Binary CFE Algorithm avoids costly inversions and multiplications.

We defined two prefixfree, complete codes $\mathrm{C_I}$, $\mathrm{C_{II}}$ mapping the PDs from $\mathbb{N}$ to $\{0,1\}^*$ and following the Gauß-Kuz'min measure. The codes $\mathrm{C_I}$, $\mathrm{C_{II}}$ yield a monotonic selfmap C on A^ω from the real interval $[0, 1)$ to the space of PD encodings.

We have seen that all three measures and induced isometries on A^ω differ pairwise and thus all three measures should be used to check a given sequence, preferably via the isometric setting provided in [14].

Acknowledgements

We thank the anonymous referees for their valuable and detailed suggestions.

References

[1] E. Berlekamp, *Non-binary BCH decoding*. TR North Carolina State University. Dept. of Statistics, 1966.

[2] M. del P. Canales Chacón, M. Vielhaber, *Structural and Computational Complexity of Isometries and their Shift Commutators*, Electronic Colloquium on Computational Complexity, ECCC TR04–057, 2004.

[3] J. L. Dornstetter. On the equivalence between Berlekamp's and Euclid's algorithm. *IEEE Trans IT*, 33(3):428–431, 1987.

[4] A. Klapper, M. Goresky. Cryptanalysis Based on 2-Adic Rational Approximation. *Crypto '95, LNCS*, 963:262–273, 1995.

[5] A. Klapper, M. Goresky. Feedback shift registers, 2-adic span, and combiners with memory. *J Crypt*, 10:111–147, 1997.

[6] R. O. Kuzmin. Sur une problème de Gauss. *Atti Congr Int Bologna*, 6:83–89, 1928.

[7] P. Lévy. Sur les lois de probabilité dont dépendent les quotients complets et incomplets d'une fraction continue *Bull. de la S.M.F*, 57:178–194, 1929.

[8] J. Massey, *Shift-register synthesis and BCH decoding.* IEEE Trans IT, 15(1), 122-127, 1969.

[9] H. Niederreiter, M. Vielhaber. *Simultaneous shifted continued fraction expansions in quadratic time.* AAECC, 9(2), 125-138, 1998.

[10] H. Niederreiter, M. Vielhaber. Tree complexity and a doubly exponential gap between structured and random sequences. *J Complexity*, 12(3):, 187–198, 1996.

[11] O. Perron. Die Lehre von den Kettenbrüchen, Bd. I. *Teubner, Stuttgart 1954/1977.*

[12] M. Vielhaber. The Karlsruhe RSA Co-processor: ISDN Network Security by RSA Encryption. *E.I.S.S. Report 89/14a*, European Institute for System Security, 1990.

[13] M. Vielhaber. 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.

[14] M. Vielhaber. A Unified View on Sequence Complexity Measures as Isometries. *SETA 2004, LNCS*, 3486:143–153, 2004.

[15] M. Vielhaber. Reduce-by-Feedback: Timing resistant and DPA-aware Modular Multiplication, plus: How to Break RSA by DPA. *CHES 2012*, Springer, 2012.

[16] M. Vielhaber. V Tree — Continued Fraction Expansion, Stern-Brocot Tree, Minkowski's $?(\mathbf{x})$ Function In Binary: Exponentially Faster. *arXiv*, 2008.08020, 2020.