ARTICLE IN PRESS
YINCO:104746

Information and Computation ••• (••••) ••••••

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Computing absolutely normal numbers in nearly linear time

Jack H. Lutz [a,*], Elvira Mayordomo [b,*]

[a] Department of Computer Science, Iowa State University, Ames, IA 50011, USA
[b] Departamento de Informática e Ingeniería de Sistemas, Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, 50018 Zaragoza, Spain

ARTICLE INFO

Article history:
Received 17 July 2020
Received in revised form 22 March 2021
Accepted 18 April 2021
Available online xxxx

Keywords:
Algorithms
Computational complexity
Lempel-Ziv parsing
Martingales
Normal numbers

ABSTRACT

A real number x is *absolutely normal* if, for every base $b \geq 2$, every two equally long strings of digits appear with equal asymptotic frequency in the base-b expansion of x. This paper presents an explicit algorithm that generates the binary expansion of an absolutely normal number x, with the nth bit of x appearing after $n$polylog(n) computation steps. This speed is achieved by simultaneously computing and diagonalizing against a martingale that incorporates Lempel-Ziv parsing algorithms in all bases.

1. Introduction

In 1909 Borel[1] [6] defined a real number α to be *normal* in base b ($b \geq 2$) if, for every $m \geq 1$ and every length-m sequence w of base-b digits, the asymptotic, empirical frequency of w in the base-b expansion of α is b^{-m}. Borel defined α to be *absolutely normal* if it is normal in every base $b \geq 2$. (This clearly anticipated the fact, proven a half-century later, that a real number may be normal in one base but not in another [10,27].) Borel's proof that *almost every* real number (i.e., every real number outside a set of Lebesgue measure 0) is absolutely normal was an important milestone in the prehistory of Kolmogorov's development of the rigorous, measure-theoretic foundations of probability theory [20]. For example, it is section 1 of Billingsley's influential textbook [5]. The recent book [9] provides a good exposition of the many aspects of current research on normal numbers.

Borel's proof shows that absolutely normal numbers are commonplace, i.e., that a "randomly chosen" real number is absolutely normal with probability 1. Rational numbers cannot be normal in even a single base b, since their base-b expansions are eventually periodic, but computer analyses of the expansions of π, e, $\sqrt{2}$, $\ln 2$, and other irrational numbers that arise in common mathematical practice suggest that these numbers are absolutely normal [7]. Nevertheless, no such "natural" example of a real number has been proven to be normal in *any* base, let alone absolutely normal. The conjectures that every algebraic irrational is absolutely normal and that π is absolutely normal are especially well known open problems [7,9,33].

* Corresponding authors.
E-mail addresses: lutz@cs.iastate.edu (J.H. Lutz), elvira@unizar.es (E. Mayordomo).

[1] Borel's original definition was that x is normal in base b if x is simply normal in all bases $b, b^2 b^3, \ldots$ Our definition here is well known to be equivalent to Borel's.

https://doi.org/10.1016/j.ic.2021.104746

The work reported here concerns a much newer problem, namely, the *complexity* of explicitly computing a real number that is provably absolutely normal, even if it is not natural in the above informal sense. Sierpinski and Lebesgue gave explicit constructions of absolutely normal numbers in 1917 [29,21], but these were intricate limiting processes that offered no complexity analyses (coming two decades before the theory of computing) and little insight into the nature of the numbers constructed. In a 1936 note that was not published in his lifetime, Turing [31] gave a constructive proof that almost all real numbers are absolutely normal and then *derived* constructions of absolutely normal numbers from this proof. Moreover, although Turing does not mention Turing machines or computability, the note is typed, with equations handwritten by him, on the back of a draft of his paper on computable real numbers [30], so it is reasonable to interpret "constructively" in a computability-theoretic sense. And in fact his proof, with 2007 corrections by Becher, Figueira, and Picchi [3], explicitly computes an absolutely normal number x. However, this algorithm is very inefficient, requiring a number of steps that is doubly exponential in n to compute the nth bit of x. (Independently, Knuth [19] published in 1965 an explicit construction of an absolutely normal number, also inefficient).

Some 75 years passed between Turing's algorithm and more efficient ones. It was only in 2013 that Becher, Heiber, and Slaman [4] published an algorithm that computes an absolutely normal number in polynomial time. Specifically, this algorithm computes the binary expansion of an absolutely normal number x, with the nth bit of x appearing after $O(n^2 f(n))$ steps for any computable unbounded nondecreasing function f. (Unpublished polynomial-time algorithms for computing absolutely normal numbers were announced independently by Mayordomo [26] and Figueira and Nies [14,15] at about the same time.) We omit here extensive work on the discrepancy, that is, the order of convergence to normality of an absolutely normal number and its tradeoff with the time complexity of the construction of the corresponding number (see the latest results in [1], [25] and their references).

In this paper we present a new algorithm that provably computes an absolutely normal in nearly linear time. Our algorithm computes the binary expansion of an absolutely normal number x, with the first to nth bits of x appearing after $O(n\text{polylog}n)$ steps. The term "nearly linear time" was introduced by Gurevich and Shelah [16]. In that paper they showed that, while linear time computability is very model-dependent, nearly linear time is very robust. For example, they showed that random access machines, Kolmogorov-Uspensky machines, Schoenhage machines, and random-access Turing machines share exactly the same notion of nearly linear time.

The novelty of our algorithm is its use of the Lempel-Ziv parsing algorithm to achieve its nearly linear time bound. For each base $b \geq 2$, we use a martingale (betting strategy) that employs the Lempel-Ziv parsing algorithm and is implicit in the work of Feder [13]. This base-b Lempel-Ziv martingale succeeds exponentially when betting on the successive digits of the base-b expansion of any real number that is not normal in base b. Our algorithm simultaneously computes and diagonalizes against (limits the winnings of) a martingale that incorporates efficient proxies of all these martingales, thereby efficiently computing a real number that is normal in every base.

The structure of this paper is based on the main result proof. Building on the base-b normality characterization in terms of finite state martingales and the universality of Lempel-Ziv martingale, we need to construct a conservative version of the Lempel-Ziv martingale that does not fluctuate too much and then establish a base change method for such conservative martingales. Finally a careful efficient combination of all resulting strategies for different bases is needed.

The rest of this paper is organized as follows. Section 2 presents the base-b Lempel-Ziv martingales and their main properties. Section 3 shows how to transform a base-b Lempel-Ziv martingale into a base-b supermartingale with an efficiently computable nondecreasing savings account that is unbounded whenever the base-b Lempel-Ziv martingale succeeds exponentially. Section 4 develops an efficient method for converting a base-b martingale with an efficiently computable savings account to a base-2 martingale that succeeds whenever the base b savings account is unbounded. Section 5 presents an algorithm that exploits the uniformity of these constructions to efficiently and simultaneously compute (a) a single base-2 martingale d that succeeds on the binary expansion of every real number x for which some base-b martingale succeeds on the base-b expansion of x and (b) a particular real number x on which binary expansion d does not succeed. This x is, perforce, absolutely normal. Section 6 presents an open problem related to our work.

For our complexity arguments, we use a (log-cost) RAM model. By [2] a nearly linear time bound on this model is equivalent a nearly linear time bound for any of the robust models in [16].

2. Lempel-Ziv martingales

For each base $b \geq 1$ we let $\Sigma_b = \{0, 1, \dots, b-1\}$ be the alphabet of *base-b digits*. We write Σ_b^* for the set of all (finite) *strings* over Σ_b and Σ_b^∞ for the set of all (infinite) *sequences* over Σ_b. We write $|x|$ for the length of a string or sequence x, and we write λ for the *empty string*, the string of length 0. For $x \in \Sigma_b^* \cup \Sigma_b^\infty$ and $0 \leq i \leq j < |x|$, we write $x[i..j]$ for the string consisting of the ith through jth digits in x. For $x \in \Sigma_b^* \cup \Sigma_b^\infty$ and $0 \leq n < |x|$, we write $x \upharpoonright n = x[0..n-1]$. For $w \in \Sigma_b^*$ and $x \in \Sigma_b^* \cup \Sigma_b^\infty$, we say that w is a *prefix* of x, and we write $w \sqsubseteq x$, if $x \upharpoonright |w| = w$.

In our (log-cost) RAM model, if an input is represented as a string $b_1 \dots b_k$ the algorithm can in time $O(\log(i))$ request and obtain the value of b_i for any i.

Let $\mathcal{D}$ be the set of dyadic rationals. Let $f : \Sigma_b^* \to [0, \infty)$ be a function. f is *nearly linear time computable* if there exists $a, c > 1$ and $\widehat{f} : \Sigma_b^* \to \mathcal{D}$ such that $\widehat{f}$ is exactly nearly linear time computable and for all $w \in \Sigma_b^*$, $|\widehat{f}(w) - f(w)| \leq a/|w|^c$. (We denote $\widehat{f}$ as a nearly linear time computation of f.)

Let $f : \Sigma_b^* \to [0, \infty)$ be a function. f is *online nearly linear time computable* if there exists $g : \Sigma_b^* \to [0, \infty)$ such that

1. f, g are nearly linear time computable, with $\widehat{f}$ and $\widehat{g}$ nearly linear time computations of them;
2. for $w \in \Sigma_b^*, a \in \Sigma_b$, $f(wa)$ and $g(wa)$ can be computed in polylogarithmic time from $w, \widehat{f}(w), \widehat{g}(w)$.

A (*base-b*) *martingale* is a function $d : \Sigma_b^* \to [0, \infty)$ satisfying

$$d(w) = \frac{1}{b} \sum_{a \in \Sigma_b} d(wa) \tag{1}$$

for all $w \in \Sigma_b^*$. (This is the original martingale notion introduced by Ville [32] and implicit in earlier papers of Lévy [23,24]. Its relationship to Doob's subsequent modifications [12], which are the "martingales" of probability theory, is explained in [18] along with the reason why Ville's original notion is still essential for algorithmic information theory.) Intuitively, a base-b martingale d is a *strategy for betting* on the subsequent digits in a sequence $S \in \Sigma_b^\infty$, with the strategy encoded in such a way that $d(S \upharpoonright n)$ is the amount of money that a gambler using the strategy d has after the first n bets. The condition (1) says that the payoffs for these bets are *fair* in the sense that the conditional expectation of $d(wa)$, given that w has occurred (and assuming that the digits $a \in \Sigma_b$ are equally likely), is $d(w)$.

A function $g : \Sigma_b^* \to [0, \infty)$ (which may or may not be a martingale) *succeeds* on a sequence $S \in \Sigma_b^\infty$ if

$$\limsup_{n \to \infty} g(S \upharpoonright n) = \infty, \tag{2}$$

i.e., if its winnings on S are unbounded. The *success set* of a function $g : \Sigma_b^* \to [0, \infty)$ is

$$S^\infty[g] = \left\{ S \in \Sigma_b^\infty \mid g \text{ succeeds on } S \right\}.$$

A function $g : \Sigma_b^* \to [0, \infty)$ *succeeds exponentially* on a sequence $S \in \Sigma_b^\infty$ if

$$\limsup_{n \to \infty} \frac{\log g(S \upharpoonright n)}{n} > 0, \tag{3}$$

i.e., if its winnings on S grow at some exponential rate, perhaps with recurrent setbacks. The *exponential success set* of a function $g : \Sigma_b^* \to [0, \infty)$ is

$$S^{\exp}[g] = \left\{ S \in \Sigma_b^\infty \mid g \text{ succeeds exponentially on } S \right\}.$$

The $f(n)$ *success set* of a function $g : \Sigma_b^* \to [0, \infty)$ is

$$S^{f(n)}[g] = \left\{ S \in \Sigma_b^\infty \,\middle|\, \limsup_{n \to \infty} \frac{\log g(S \upharpoonright n)}{\log f(n)} \geq 1 \right\}.$$

Note that $S^{\exp}[g] = \cup_{\epsilon > 0} S^{2^{\epsilon n}}[g]$.

For technical reasons we will also need to consider the notion of supermartingale, which in many contexts turns out to be equivalent to the notion of martingale.

A (*base-b*) *supermartingale* is a function $d : \Sigma_b^* \to [0, \infty)$ satisfying

$$d(w) \leq \frac{1}{b} \sum_{a \in \Sigma_b} d(wa) \tag{4}$$

for all $w \in \Sigma_b^*$.

Lemma 2.1. *For each online nearly linear time computable supermartingale d there is an online nearly linear time computable martingale d' and a constant $\alpha > 0$ such that for every $w \in \Sigma_b^*$ $d'(w) \geq d(w) - \alpha$. If for every $w \in \Sigma_b^*$ $d(w) \leq a|w|^c$, then $d'(w) \leq a|w|^{c+1} - \alpha$. If for some m, C, $d(w) = C$ for $|w| \leq m$ then $d'(w) = C$ for $|w| \leq m$.*

Proof (proof sketch). We define d' recursively as follows, $d'(\lambda) = d(\lambda)$ and for $w \in \Sigma_b^*, a \in \Sigma_b$

$$d'(wa) = d'(w) + \widehat{d}(wa) - 1/b \sum_{b'} \widehat{d}(wb')$$

d' is online nearly linear computable. It can be proven by induction that d' is a martingale and that for all for $w \in \Sigma_b^*$, $d'(w) \geq \widehat{d}(w)$ and $d'(w) \leq \sum_{u \sqsubseteq w} \widehat{d}(u)$. $\square$

A function $g : \Sigma_b^* \to [0, \infty)$ *succeeds strongly* on a sequence $S \in \Sigma_b^\infty$ if (2) holds with the limit superior replaced by a limit inferior i.e., if the winnings converge to ∞. A function $g : \Sigma_b^* \to [0, \infty)$ *succeeds strongly exponentially* on a sequence

```
input  w ∈ Σ_b^* ;
x, L(x), d = λ, 1, 1 ;
T, j = {λ}, 0 ;
while true do
begin
    if w = λ then output d and halt ;
    if L(x) = 1 then
    begin
        L(x) = b ;
        for each 0 ≤ a < b do L(xa) = 1 ;
        T, x(j), j = T ∪ {x}Σ_b, x, j + 1 ;
        x = λ ;
    end
    else
    begin
        a, w = head(w), tail(w) ;
        L(x), x, d = L(x) + b − 1, xa, (bL(xa)/L(x)) d
    end
end
```

Fig. 1. Algorithm for computing $d_{\mathrm{LZ}(b)}(w)$.

$S \in \Sigma_b^\infty$ if (3) holds with the limit superior replaced by a limit inferior i.e., if the winnings grow at exponential rate. The *strong success sets* $S_{\mathrm{str}}^{\infty}[g]$ and $S_{\mathrm{str}}^{\mathrm{exp}}[g]$ of a function $g : \Sigma_b^* \to [0, \infty)$ are defined in the now-obvious manner. It is clear that the inclusions

$$S_{\mathrm{str}}^{\mathrm{exp}}[g] \subseteq S^{\mathrm{exp}}[g] \subseteq S^{\infty}[g]$$

and

$$S_{\mathrm{str}}^{\mathrm{exp}}[g] \subseteq S_{\mathrm{str}}^{\infty}[g] \subseteq S^{\infty}[g]$$

hold for all $g : \Sigma_b^* \to [0, \infty)$.

For each base $b \geq 2$ the *base-b Lempel-Ziv martingale* is a particular martingale $d_{\mathrm{LZ}(b)}$ based on the Lempel-Ziv parsing algorithm [22], as we now explain.

Formally, $d_{\mathrm{LZ}(b)}$ is computed by the algorithm in Fig. 1, but some explanation is appropriate here. The algorithm is written with several instances of parallel assignment. For example, the second line initializes x, $L(x)$, and d to the values λ, 1, and 1, respectively. The items T, j, and $x(j)$ are not needed for the computation of $d_{\mathrm{LZ}(b)}(w)$, but they are useful for understanding and analyzing the algorithm.

The growing set T of strings in Σ_b^* always contains all the prefixes of all its elements, so it is a tree. We envision this tree as being oriented with its root at the top and the immediate children $v0, v1, \ldots, v(b-1)$ of each interior vertex v of T displayed left-to-right below T. The *dictionary* of the algorithm is the current set of leaves of T.

The string x in the algorithm is always an element of (i.e., location in) the tree T, and $L(x)$ is always the number of leaves of T that are descendants of x. We regard x as a descendant of itself, so x is a leaf if and only if $L(x) = 1$.

It is clear that $d_{\mathrm{LZ}(b)}(\lambda) = 1$. In fact, the algorithm's successive values of d are the values $d_{\mathrm{LZ}(b)}(u)$ for successive prefixes u of the input string w. More precisely, if w_t and d_t are the values of w and d after t executions of the else-block, then $w = (w \upharpoonright t)w_t$ and $d_t = d_{\mathrm{LZ}(b)}(w \upharpoonright t)$.

For $w \in \Sigma_b^*$ we define the tree $T(w)$ as follows. If $w = \lambda$, then $T(w) = \{\lambda\}$. If $w = w'a$, where $w' \in \Sigma_b^*$ and $a \in \Sigma_b$, then $T(w'a)$ is the value of T when the algorithm terminates on input w'. (Note that this is one step before it terminates on input $w'a$.) For $w \in \Sigma_b^*$ we define $D(w)$ to be the number of leaves in $T(w)$. For each x in $T(w)$, $L(x, w)$ is the number of leaves of $T(w)$ that are descendants of x.

The computation is divided into "epochs". At the beginning of each epoch, the string x is λ, i.e., it is located at the root of T. The string x then takes successive digits from whatever is left of w (because $a, w = \mathrm{head}(w), \mathrm{tail}(w)$ removes the first digit of w and stores it in a), following this path down the tree and updating d at each step, until w is empty (the end of the last epoch) or x is a leaf of T. In the latter case, the jth epoch is over, the b children $x0, x1, \ldots, x(b-1)$ are added to T as new leaves, x is the jth *phrase* $x(j)$ of w, and x is reset to the root λ of T.

When the algorithm terminates, it is clear that exactly one of the following things must hold.

(a) $w = \lambda$.
(b) $w = x(0) \ldots x(j-1)$.
(c) $w = x(0) \ldots x(j-1)u$ for some nonempty interior vertex u of $T(w)$.

In case (a) or (b) we call w a *full parse*. In case (b) or (c) we call $x(0), \ldots, x(j-1)$ the *full phrases* of w. In case (c) we call u the *partial phrase* of w.

Notice that the function $h(w) = (T(w), j(w), D(w))$ is online nearly linear time computable. Notice that our algorithm does not work in nearly linear time when computing d as a product of $|w|$ factors. We give below an alternative characterization of $d_{\mathrm{LZ}(b)}$ that will be useful later.

Define the set $A_b = \left\{1 + (b-1)r \,\middle|\, r \in \mathbb{N}\right\}$ and the generalized factorial function $fact_b : A_b \to A_b$ by

$$fact_b(1 + (b-1)r) = \prod_{k=1}^{r}(1 + (b-1)k)$$

for all $r \in \mathbb{N}$.

Observation 2.2. *For all $n \in A_b$,*

$$1 \leq \frac{fact_b(n)}{e^{\frac{1}{b-1}}\left(\frac{n}{e}\right)^{\frac{n}{b-1}}} \leq n. \tag{5}$$

Using Euler-Maclaurin formula we also have

Observation 2.3. *For all $n \in A_b$,*

$$fact_b(n) = \tag{6}$$
$$C \cdot n^{\frac{n}{b-1}} \cdot e^{-(n-1)(b-1)} \cdot n^{1/2} \cdot e^{(b-1)/(12n)} \cdot e^{-(b-1)^3/(720n^3)} \cdot e^{(b-1)^5/(30240n^5)} \cdot e^{O(1/n^7)}. \tag{7}$$

Using the terms in Observation 2.3 we define

$$\widehat{fact_b}(n) = C \cdot n^{\frac{n}{b-1}} \cdot e^{-(n-1)(b-1)} \cdot n^{1/2} \cdot e^{(b-1)/(12n)} \cdot e^{-(b-1)^3/(720n^3)} \cdot e^{(b-1)^5/(30240n^5)}. \tag{8}$$

All terms in the definition of $\widehat{fact_b}(n)$ are computed with an approximation of $e^{O(1/n^7)}$, that is, all terms in the exponents will have precision $7\log n + O(1)$.

Lemma 2.4. *(Feder [13]) Let $w \in \Sigma_b^*$.*

1. *If w is a full parse, then*

$$d_{\mathrm{LZ}(b)}(w) = \frac{b^{|w|}}{fact_b(D(w))}.$$

2. *If w is not a full parse and u is its partial phrase, then*

$$d_{\mathrm{LZ}(b)}(w) = \frac{b^{|w|}}{fact_b(D(w))}L(u, w),$$

where $L(u, w)$ is the number of leaves below u in $T(w)$.

The following lemma follows from Lemma 2.4.

Lemma 2.5. *For $S \in \Sigma_b^\infty$ and $\alpha \in (0, 1)$ the following three conditions are equivalent.*

(a) $S \in S^{b^{(1-\alpha)n}}[d_{\mathrm{LZ}(b)}]$.
(b) *There exist infinitely many full parses $w \sqsubseteq S$ for which*

$$D(w)\log_b|w| < \alpha(b-1)|w|.$$

(c) *There exist infinitely many full parses $w \sqsubseteq S$ for which*

$$D(w)\log_b D(w) < \alpha(b-1)|w|.$$

Corollary 2.6. *For $S \in \Sigma_b^\infty$ the following three conditions are equivalent.*

(a) $S \in S^{\exp}[d_{\mathrm{LZ}(b)}]$.

(b) *There exist* $\alpha < 1$ *and infinitely many full parses* $w \sqsubseteq S$ *for which*

$$D(w)\log_b |w| < \alpha(b-1)|w|.$$

(c) *There exist* $\alpha < 1$ *and infinitely many full parses* $w \sqsubseteq S$ *for which*

$$D(w)\log_b D(w) < \alpha(b-1)|w|.$$

We conclude this section by explaining the connection between the Lempel-Ziv martingales and normality. First, Schnorr and Stimm [28] defined (implicitly) a notion of *finite-state base-b martingale* and proved that every sequence $S \in \Sigma_b^\infty$ obeys the following dichotomy.

1. If S is normal, then no finite-state base-b martingale succeeds on S. (In fact, every finite-state base-b martingale decays exponentially on S.)
2. If S is not normal, then some finite-state base-b martingale succeeds exponentially on S.

Some twenty years later, Feder [13] defined (implicitly) the Lempel-Ziv martingale $d_{\mathrm{LZ}(b)}$ and proved (implicitly) that $d_{\mathrm{LZ}(b)}$ is at least as successful *on every sequence* as every finite-state base-b martingale. That is, for finite-state base-b martingale d, the inclusions

$$S^\infty[d] \subseteq S^\infty[d_{\mathrm{LZ}(b)}],\; S^\infty_{\mathrm{str}}[d] \subseteq S^\infty_{\mathrm{str}}[d_{\mathrm{LZ}(b)}],$$
$$S^{\mathrm{exp}}[d] \subseteq S^{\mathrm{exp}}[d_{\mathrm{LZ}(b)}],\; S^{\mathrm{exp}}_{\mathrm{str}}[d] \subseteq S^{\mathrm{exp}}_{\mathrm{str}}[d_{\mathrm{LZ}(b)}]$$

all hold. This, together with Schnorr and Stimm's dichotomy result, implies that $d_{\mathrm{LZ}(b)}$ succeeds exponentially on every non-normal sequence in Σ_b^∞. Hence a real number x is absolutely normal if none of the martingales $d_{\mathrm{LZ}(b)}$ succeed exponentially on the base-b expansion of x.

In order to avoid time bounds that are dependent on the alphabet size b, we will consider the following variant of $d_{\mathrm{LZ}(b)}$,

$$d_{\mathrm{LZ+}(b)}(w) = \begin{cases} 1, & \text{if } |w| \leq 2^b \\ \frac{d_{\mathrm{LZ}(b)}(w)}{d_{\mathrm{LZ}(b)}(w \upharpoonright 2^b)}, & \text{if } |w| > 2^b. \end{cases}$$

Notice that $S^\infty[d_{\mathrm{LZ}(b)}] = S^\infty[d_{\mathrm{LZ+}(b)}]$ and that for any a.e. unbounded f, $S^{f(n)}[d_{\mathrm{LZ}(b)}] = S^{f(n)}[d_{\mathrm{LZ+}(b)}]$.

Notice that if $\log(|w|) > b$ then a polynomial bound on b is a polylogarithmic bound on $|w|$.

3. Savings accounts

In this section we construct a conservative version of the Lempel-Ziv martingale $d_{\mathrm{LZ+}(b)}$ consisting of a new supermartingale d' that can be smaller than $d_{\mathrm{LZ+}(b)}$ but that has a savings account in the sense explained next. We will need this conservative version in the base change transformation in section 4.

Definition. *A function* $g : \Sigma_b^* \to [0,\infty)$ *is a savings account of a supermartingale* $d : \Sigma_b^* \to [0,\infty)$ *if g is nondecreasing with respect to substring order and, for every* $w \in \Sigma_b^*$, $d(w) \geq g(w)$.

In the following construction we use Observation 2.2 and Lemma 2.4 to get a far more conservative version of $d_{\mathrm{LZ+}(b)}$. We define a function $goal(w)$ such that

$$b \leq d_{\mathrm{LZ+}(b)}(w) b^{-goal(w)} \leq b^6 |w|^2$$

and then a nondecreasing upper bound $taken(w) \geq goal(w)$ such that for every S, $taken(S \upharpoonright n)$ coincides with $goal(S \upharpoonright n)$ infinitely often.

Construction 3.1. *Let* $d = d_{\mathrm{LZ+}(b)}$ *be the base-b-Lempel-Ziv martingale. We define a new supermartingale* $d' = e' + g'$ *as follows.*

We first define e'. *Let* $w \in \Sigma_b^*$. *For* $|w| \leq 2^b$, $e'(w) = b$, $taken(w) = 0$. *For* $|w| > 2^b$, *let* $w = x(0)\dots x(j-1)u$, *for* $z = x(0)\dots x(j-1)$ *a full parse and u the partial or the last full phrase of w. Let*

$$goal(w) = |w| - \lceil D(w)(\log_b(D(w)))/(b-1)\rceil + \lfloor D(w)(\log_b(e))/(b-1)\rfloor$$
$$- \lceil \log_b(D(w))\rceil - \lceil \log_b e^{\frac{1}{b-1}}\rceil - \lceil \log_b d_{\mathrm{LZ}(b)}(w \upharpoonright 2^b)\rceil - 1,$$
$$taken(w) = \max\{taken(z), goal(w)\},$$
$$e'(w) = d(w) b^{-taken(w)}.$$

Let g' be defined as follows. Let $w \in \Sigma_b^*$, $a \in \Sigma_b$. For $|w| \leq 2^b$, $g'(w) = 0$ and for $|w| \geq 2^b$,

$$g'(wa) = \begin{cases} g'(w) & \text{if } goal(wa) \leq taken(w) \\ g'(w) + e'(w)\frac{b-1}{b} & \text{if } goal(wa) > taken(w) \end{cases}$$

Theorem 3.2. *Let d' and g' be as defined in Construction 3.1. Then d' is a supermartingale and g' is its savings account,*

$$S^{\exp}[d_{LZ(b)}] \subseteq S^{\infty}[g'],$$

d' is computable in an online nearly linear time bound that does not depend on b, and there exists $a, c > 1$ not depending on b such that for every $w \in \Sigma_b^+$, $d'(w) \leq b \cdot a \cdot |w|^c$, $d'(\lambda) = b$.

Proof of Theorem 3.2.

Notice that from Construction 3.1, we can assume that $b < \log|w|$ for all complexity bounds in this proof.

Claim 3.3. *$d' = e' + g'$ is a supermartingale and g' is a savings account for d'.*

Proof. Let us prove that d' is a supermartingale. Note that by definition of D, $goal(wa)$ does not depend on a. When $goal(wa) \leq taken(w)$ we have that

$$\sum_{a \in \Sigma_b} e'(wa) = \frac{e'(w)}{d(w)} \sum_{a \in \Sigma_b} d(wa) = b \cdot \frac{e'(w)}{d(w)} \cdot d(w) = b \cdot e'(w).$$

Since $g'(wa)$ is constant the martingale equality holds in this case.

In the second case, when $goal(wa) > taken(w)$, since they are integer values $goal(wa) \geq taken(w) + 1$. We have that

$$\begin{aligned} \sum_{a \in \Sigma_b} d'(wa) &= \textstyle\sum_{a \in \Sigma_b} e'(wa) + \sum_{a \in \Sigma_b} g'(wa) \\ &= \tfrac{e'(w)}{d(w)} \textstyle\sum_{a \in \Sigma_b} b^{-goal(wa)+taken(w)} d(wa) + \sum_{a \in \Sigma_b} g'(wa) \\ &\leq e'(w) + b(g'(w) + e'(w)\tfrac{b-1}{b}) = b(e'(w) + g'(w)) = bd'(w). \end{aligned}$$

Since e' is nonnegative, by definition g' is a nondecreasing function. Therefore g' is a savings account of d'. □

Claim 3.4. *For every $w \in \Sigma_b^*$ with $|w| > 2^b$*

$$e'(w) = d(w)b^{-taken(w)} \leq b^6 \cdot D(w) \cdot L(u, w).$$

$$d(w)b^{-goal(w)} \geq b$$

for u the partial or the last full phrase of w.

Proof. Use that $taken(w) \geq goal(w)$, Lemma 2.4, and Observation 2.2. □

Claim 3.5. *If $y \in \Sigma_b^\infty$ and $goal(y \upharpoonright n)$ is unbounded then $y \in S^\infty[g']$.*

Proof. If $goal(y \upharpoonright n)$ is unbounded then infinitely often we use the second case in the definition of g' and have that $taken(y \upharpoonright n) = goal(y \upharpoonright n) > taken(y \upharpoonright (n-1))$, $e'(y \upharpoonright n) = d(y \upharpoonright n)b^{-goal(y \upharpoonright n)}$, and $g'(y \upharpoonright n) = g'(y \upharpoonright (n-1)) + e'(y \upharpoonright (n-1))\frac{b-1}{b}$.

Since $goal(y \upharpoonright (n-1)) \geq goal(y \upharpoonright n) - 1$, then $taken(y \upharpoonright (n-1)) = goal(y \upharpoonright (n-1))$. By Claim 3.4, $e'(y \upharpoonright (n-1)) \geq b$, therefore $g'(y \upharpoonright n) \geq g'(y \upharpoonright (n-1)) + b - 1$.

Since g' is monotonic, $y \in S^\infty[g']$. □

Claim 3.6. *For every $\alpha \in (0, 1)$, $S^{b^{(1-\alpha)n}}[d_{LZ(b)}] \subseteq S^\infty[g']$.*

Proof. If $y \in S^{b^{(1-\alpha)n}}[d_{LZ(b)}]$ then by Lemma 2.5 for infinitely many n, $D(y \upharpoonright n)\log_b(D(y \upharpoonright n)) < \alpha(b-1)n$.

Notice that therefore $goal(y \upharpoonright n)$ is unbounded and by Claim 3.5 $y \in S^\infty[g']$. □

Claim 3.7. *For every $w \in \Sigma_b^*$, $e'(w) \leq e'(\lambda)\log(|w|)^6|w|^2$ and $g'(w) \leq \sum_{v \sqsubseteq w} e'(v)$. Therefore $d'(w)$ is polynomially bounded.*

Proof. Use Claim 3.4. □

Claim 3.8. *d' can be computed in online nearly linear time (in time $n\log^c n$ for c not depending on b).*

Proof. We first proof that $e' : \Sigma_b^* \to [0,\infty)$ can be computed in online nearly linear time. Let

$$\widehat{e'}(w) = \frac{b^{|w|}}{\widehat{fact_b}(D(w))} L(u,w) b^{-taken(w)}.$$

Then

$$\begin{aligned}|e'(w) - \widehat{e'}(w)| &\le b^{|w|-taken(w)} L(u,w) \left| \frac{1}{fact_b(D(w))} - \frac{1}{\widehat{fact_b}(D(w))} \right| \\ &\le b^{|w|-taken(w)} L(u,w) \left| \frac{\widehat{fact_b}(D(w)) - fact_b(D(w))}{fact_b(D(w)) \cdot \widehat{fact_b}(D(w))} \right| \\ &\le b^{|w|-taken(w)} L(u,w) \left| \frac{1 - e^{O(1/D(w)^7)}}{fact_b(D(w))} \right| = e'(w)|1 - e^{O(1/D(w)^7)}| \\ &\le e'(w) \frac{1}{CD(w)^7} + O(1/D(w)^{14}) \\ &\le \frac{1}{CD(w)^4} + O(1/D(w)^{11})\end{aligned}$$

Notice that $D(w) \ge \sqrt{|w|}$, therefore,

$$|e'(w) - \widehat{e'}(w)| \le b/C1/|w|^2 + O(1/|w|^5).$$

Since $\widehat{e'}$ is computable in online nearly linear time, e' is too.

Again, for $w \in \Sigma_b^*$, let $w = x(0)\dots x(j-1)u$, for $z = x(0)\dots x(j-1)$ a full parse and u a partial or full phrase of w. If $goal(w) > taken(z)$, let t be the shortest such that $t \sqsubseteq u$ and $goal(zt) > taken(z)$.

Notice that by Lemma 2.4 and the definition of g',

$$g'(w) = \begin{cases} g'(z) & \text{if } goal(w) \le taken(z) \\ g'(z) + \sum_{v \sqsubseteq u, |zv| \ge |zt|} e'(zt)/L(t,w) \frac{b-1}{b} L(v,w) & \text{if } goal(w) > taken(z), \end{cases}$$

where $L(v,w)$ is the number of leaves below v in $T(w)$. Given precomputed values for $f(u) = \sum_{v \sqsubseteq u} L(v,w)$, the value of $g'(w)$ can be easily computed in online nearly linear time. □

This completes the proof of Theorem 3.2. □

4. Base change

We use infinite sequences over Σ_b to represent real numbers in $[0,1)$. For this, we associate each string $w \in \Sigma_b^*$ with the half-open interval $[w]_b$ defined by $[w]_b = [x, x + b^{-|w|})$, for $x = \sum_{i=1}^{|w|} w[i-1]b^{-i}$. Each real number $\alpha \in [0,1)$ is then represented by the unique sequence $\mathrm{seq}_b(\alpha) \in \Sigma_b^\infty$ satisfying

$$w \sqsubseteq \mathrm{seq}_b(\alpha) \iff \alpha \in [w]_b$$

for all $w \in \Sigma_b^*$. We have

$$\alpha = \sum_{i=1}^{\infty} \mathrm{seq}_b(\alpha)[i-1]b^{-i}$$

and the mapping $\mathrm{seq}_b : [0,1) \to \Sigma_b^\infty$ is a bijection. (Notice that $[w]_b$ being half-open prevents double representations.) We define $\mathrm{real}_b : \Sigma_b^\infty \to [0,1)$ to be the inverse of seq_b. A set of real numbers $A \subseteq [0,1)$ is represented by the set

$$\mathrm{seq}_b(A) = \{\mathrm{seq}_b(\alpha) \mid \alpha \in A\}$$

of sequences. If $X \subseteq \Sigma_b^\infty$ then

$$\mathrm{real}_b(X) = \{\mathrm{real}_b(x) \mid x \in X\}.$$

Construction 4.1. *Let $d : \Sigma_b^* \to [0, \infty)$ be a polynomially-bounded martingale with a savings account g.*

We define $\gamma : \Sigma_b^* \to [0, 1]$ a probability measure on Σ_b^∞, $\gamma(w) := b^{-|w|} d(w)/d(\lambda)$.

Using the Carathéodory extension to Borel sets, γ can be extended to any interval $[a, c]$; we denote with $\widehat{\gamma}$ this extension. (In fact if we consider all $U \subseteq \Sigma_b^*$ such that all $u, v \in U, u \neq v$ are incomparable and $[u]_b \subseteq [a, c]$ for all $u \in U$, then $\widehat{\gamma}([a, c]) = \sup_U \sum_{u \in U} \gamma(u)$).

We define $\mu : \{0, 1\}^* \to [0, 1]$ by $\mu(y) = \widehat{\gamma}([y]_2)$.

Finally we define $d^{(2)} : \{0, 1\}^* \to [0, \infty)$ by $d^{(2)}(y) = 2^{|y|} \mu(y)$.

Theorem 4.2. *Let d be a base-b martingale that is polynomially bounded such that $d(w)$ is constant for $|w| \leq 2^b$, and let g be a savings account of d. Let $d^{(2)}$ be defined from d and g as in Construction 4.1. Then*

$$real_b(S^\infty[g]) - \mathbb{Q} \subseteq real_2(S^\infty[d^{(2)}]).$$

Moreover, if d is computable in an online nearly linear time bound not depending on b, then so is $d^{(2)}$. If for all $w \in \Sigma_b^+$, $d(w) \leq a|w|^c$ then for all $y \in \{0, 1\}^+$, $d^{(2)}(y) \leq 3a|y|^c/d(\lambda)$.

Proof of Theorem 4.2. We will first show that Carathéodory extension of d works for sequences base change, and then approximate the resulting $d^{(2)}$ using d restricted to Σ_b^m for a fixed m.

Property 4.3. *If for all $w \in \Sigma_b^*$, $d(w) \leq a|w|^c$ then for all $y \in \{0, 1\}^*$, $d^{(2)}(y) \leq 3a|y|^c/d(\lambda)$.*

Let $y \in \{0, 1\}^*$. Let $A_y = \left\{ w \in \Sigma_b^* \,|\, |w| = |y| \text{ and } [w]_b \cap [y]_2 \neq \emptyset \right\}$. Then

$$\begin{aligned} d^{(2)}(y) &\leq 2^{|y|} \sum_{w \in A_y} b^{-|w|} d(w)/d(\lambda) \\ &\leq 2^{|y|} \sum_{w \in A_y} b^{-|w|} a|w|^c/d(\lambda) \\ &\leq 2^{|y|}(2^{-|y|} + 2b^{-|y|}) a|y|^c/d(\lambda) \\ &\leq 3a|y|^c/d(\lambda) \end{aligned}$$

Property 4.4. *Let $\alpha \in [0, 1) - \mathbb{Q}$. If $\mathrm{seq}_b(\alpha) \in S^\infty[g]$, then $\mathrm{seq}_2(\alpha) \in S^\infty[d^{(2)}]$.*

Proof. Let $x = \mathrm{seq}_b(\alpha) \in S^\infty[g]$. Let $y = \mathrm{seq}_2(\alpha)$.

We use here that d has a savings account g, so if $g(x \upharpoonright n) > m$ then for all w with $x \upharpoonright n \sqsubseteq w$, $g(w) > m$.

Let $m \in \mathbb{N}$ and choose n such that $g(x \upharpoonright n) > m$. Let q be such that $[y \upharpoonright q]_2 \subseteq [x \upharpoonright n]_b$ (this q exists because $\alpha \in [0, 1) - \mathbb{Q}$). Let us see that $d^{(2)}(y \upharpoonright q) > m/d(\lambda)$.

Let $r \in \mathbb{N}$. Let $A_r^q = \left\{ w \in \Sigma_b^* \,|\, |w| = r \text{ and } [w]_b \subseteq [y \upharpoonright q]_2 \right\}$. Then

$$\begin{aligned} d^{(2)}(y \upharpoonright q) &= 2^q \widehat{\gamma}([y \upharpoonright q]_2) = 2^q \lim_r \sum_{w \in A_r^q} d(w)/d(\lambda) b^{-|w|} \\ &\geq 2^q m/d(\lambda) \lim_r \sum_{w \in A_r^q} b^{-|w|} = 2^q m/d(\lambda) 2^{-q} = m/d(\lambda). \end{aligned}$$

The last chain of equations holds because $[y \upharpoonright q]_2 \subseteq [x \upharpoonright n]_b$ and for every $w \in A_r^q$, $[w]_b \subseteq [y \upharpoonright q]_2$, so $x \upharpoonright n \sqsubseteq w$ for any $w \in A_r^q$. □

We next compute $d^{(2)}$. For each $m \in \mathbb{N}$ we define $\mu_m : \{0, 1\}^* \to [0, 1]$ by

$$\mu_m(y) = \sum_{|w| = m, [w]_b \cap [y]_2 \neq \emptyset} \gamma(w).$$

Claim 4.5. *For every $y \in \{0, 1\}^*$ and $m \in \mathbb{N}$, $|\mu(y) - \mu_m(y)| \leq 2b^{-m} am^c/d(\lambda)$.*

Proof. Let a, c be such that $d(w) \leq a|w|^c$ for every w (using that d is polynomially bounded). Then since at most two strings w with $|w| = m$ have the property that $[w]_b \cap [y]_2 \neq \emptyset$ and $[w]_b \not\subseteq [y]_2$, we have

$$|\mu(y) - \mu_m(y)| \leq 2b^{-m} am^c/d(\lambda). \quad \square$$

For each $m \in \mathbb{N}$ we define $d_m^{(2)} : \{0,1\}^* \to [0,\infty)$ by $d_m^{(2)}(y) = 2^{|y|}\mu_m(y)$.

Claim 4.6. *For some $c > 0$, for every $y \in \{0,1\}^*$, for every $m \in \mathbb{N}$,*

$$|d^{(2)}(y) - d_m^{(2)}(y)| \le 2^{|y|}2^{-m\log b + c\log m + 1}/d(\lambda).$$

Corollary 4.7. *For some $c' > 0$, for every $y \in \{0,1\}^*$,*

$$|d^{(2)}(y) - d^{(2)}_{|y|/\log b + c'\log|y|}(y)| \le 1/|y|^3 \cdot 1/d(\lambda).$$

Proof. Take $c' = (4+c)/\log b$ and use the previous claim. □

Property 4.8. *For $m \in \mathbb{N}$, $y \in \{0,1\}^*$, $d_m^{(2)}(y)$ can be computed by considering a maximum of $2b$ neighbor strings $w \in \Sigma_b^r$ for $r = |y|/\log b$ to m, computing $d(w)$ for each of them and doing an addition and a multiplication for each.*

Proof. Consider P, the smallest prefix free set of strings $w \in \Sigma_b^*$ such that $[w]_b \subseteq [y]_2$, and notice that $|w| \ge |y|/\log b$ for each such string. For each r there are at most $2b-2$ strings of length r in P (otherwise we can replace some of them by a single string of length $r-1$). For length m we may need two more strings $|w| = m, [w]_b \cap [y]_2 \neq \emptyset$. □

Corollary 4.9. *For $y \in \{0,1\}^*$, $d^{(2)}_{|y|/\log b + c'\log|y|}(y)$ can be computed by considering a maximum of $2b$ neighbor strings $w \in \Sigma_b^r$ for $r = |y|/\log b$ to $|y|/\log b + c'\log|y|$, computing $d(w)$ for each of them and doing an addition and a multiplication for each.*

By Corollary 4.7 $f(y) = d^{(2)}_{|y|/\log b + c'\log|y|}(y)$ approximates $d^{(2)}(y)$ within a $1/|y|^3 \cdot 1/d(\lambda)$ bound, and by the last corollary f can be computed in online nearly linear time. Using that $d(w)$ is constant for $|w| \le 2^b$ we have a nearly linear time bound independent of b.

This concludes the proof of Theorem 4.2. □

5. Absolutely normal numbers

In this section we give an algorithm that diagonalizes against the Lempel-Ziv martingales for all bases in nearly linear time.

We use the following theorem which is a union lemma for online nearly linear martingales that works for a set of martingales that is uniformly computable, uniformly approximated and uniformly polynomially bounded.

Theorem 5.1. *Let $(d_k)_{k\in\mathbb{N}}$ be a sequence of base-2 martingales such that for each of them there exists a function $\widehat{d_k} : \{0,1\}^* \to [0,\infty)$ with the following properties*

1. *$\widehat{d_k}$ is computable in an online nearly linear time bound that does not depend on k.*
2. *There is $a, c > 1$ such that for every $k \in \mathbb{N}$, $y \in \{0,1\}^*$*

$$|d_k(y) - \widehat{d_k}(y)| \le \frac{a}{|y|^c}.$$

3. *$d_k(\lambda) = \widehat{d_k}(\lambda) = 1$ and there is $a, c > 1$ such that for every $k \in \mathbb{N}$, $y \in \{0,1\}^*$, $d_k(y) \le a|y|^c$.*

Then we can compute in online nearly linear time a binary sequence x such that, for every k, $x \notin S^\infty[d_k]$.

Proof. Let $d : \{0,1\}^* \to [0,\infty)$ be defined by

$$d(w) = \sum_{k=1}^{\infty} 2^{-k} d_k(w).$$

Then d is a martingale. Let $\widehat{d} : \{0,1\}^* \to [0,\infty)$ be

$$\widehat{d}(w) = \sum_{k=1}^{(c+2)\log|w| + \log a} 2^{-k}\widehat{d_k}(w).$$

Notice that $\widehat{d}$ is computable in online nearly linear time.

Claim 5.2. *For each* $w \in \{0,1\}^*$, $|d(w) - \widehat{d}(w)| \le (a+1)/|w|^{c+1}$.

$$
\begin{aligned}
|d(w) - \widehat{d}(w)| &\le \sum_{k=1}^{(c+2)\log|w|+\log a} 2^{-k}|d_k(w) - \widehat{d_k}(w)| + \sum_{k=(c+2)\log|w|+\log a+1}^{\infty} 2^{-k} d_k(w) \\
&\le \sum_{k=1}^{(c+2)\log|w|+\log a} 2^{-k} a/|w|^c + \sum_{k=(c+2)\log|w|+\log a+1}^{\infty} 2^{-k} a|w|^c \\
&\le a/|w|^c + a|w|^c/(a|w|^{c+2}) = a/|w|^c + 1/|w|^2.
\end{aligned}
$$

Our algorithm will diagonalize against d, constructing a binary sequence x as follows. If $x \upharpoonright n$ has been defined then choose the next bit of x as $i \in \{0,1\}$ that minimizes $\widehat{d}((x \upharpoonright n)i)$.

Claim 5.3. *If* $x \notin S^\infty[d]$ *then for every* k, $x \notin S^\infty[d_k]$.

Let $n \in \mathbb{N}, k \in \mathbb{N}$, $d_k(x \upharpoonright n) \le 2^k d(x \upharpoonright n)$.

Claim 5.4. $x \notin S^\infty[d]$.

Let $w \in \{0,1\}^*$. We prove that for $i \in \{0,1\}$ chosen to minimize $\widehat{d}(wi)$ it holds that $d(wi) \le d(w) + (a+1)/(|w|+1)^{c+1}$. Since $\widehat{d}(wi) \le \widehat{d}(w(1-i))$ it holds that

$$
\begin{aligned}
d(wi) &\le \widehat{d}(wi) + (a+1)/(|w|+1)^{c+1} \\
&\le \widehat{d}(w(1-i)) + (a+1)/(|w|+1)^{c+1} \le d(w(1-i)) + 2(a+1)/(|w|+1)^{c+1}.
\end{aligned}
$$

Since d is a martingale, it follows that $d(wi) \le d(w) + (a+1)/(|w|+1)^{c+1}$.
Therefore

$$
d(x \upharpoonright n) \le d(x \upharpoonright (n-1)) + (a+1)/n^{c+1}
$$

and $x \notin S^\infty[d]$. □

We now have all the ingredients for our main result.

Theorem 5.5. *There is an explicit algorithm that computes the binary expansion of an absolutely normal number* z *in online nearly linear time.*

Proof. The algorithm arises from a combination of Theorem 3.2, Lemma 2.1, Theorem 4.2, and Theorem 5.1, notice that all of them give fully explicit constructions.

As explained in section 2, a real number z is absolutely normal if none of the martingales $d_{\mathrm{LZ}+(b)}$ succeed exponentially on the base-b expansion of z.

For each b, let d be the polynomially bounded and online nearly linear time computable supermartingale with a savings account g' defined in Theorem 3.2 and construction 3.1 as a conservative substitute of $d_{\mathrm{LZ}+(b)}$ (that is, $S^{\exp}[d_{\mathrm{LZ}(b)}] = S^{\exp}[d_{\mathrm{LZ}+(b)}] \subseteq S^\infty[g']$).

Let d' be the online nearly linear time computable and polynomially bounded martingale with $d'(w) \ge d(w)$ for all w given by Lemma 2.1. Notice that since $d'(w) \ge d(w)$, g' is a savings account for d'.

For $b \ne 2$, we now use Theorem 4.2 for d', g' and we have an online nearly linear time computable martingale $d^{(2)} : \{0,1\}^* \to [0,\infty)$ that succeeds on the base-2 expansion of the irrational reals with base-b expansion in $S^\infty[g']$ ($real_b(S^\infty[g']) - \mathbb{Q} \subseteq real_2(S^\infty[d^{(2)}])$).

For $b = 2$ we directly take $d^{(2)}(w) = d'(w)/d'(\lambda)$. Notice that $\mathbb{Q} \subseteq real_b(S^\infty[d'])$ because $\mathrm{seq}_b(\mathbb{Q}) \subseteq S^{\exp}[d_{\mathrm{LZ}(b)}]$.

For each b the computation of $d_b = d^{(2)}$ fulfills the conditions of Theorem 5.1, so we can compute in online nearly linear time a binary sequence x such that, for every b, $x \notin S^\infty[d_b]$ and therefore $real_2(x) \notin real_b(S^{\exp}[d_{\mathrm{LZ}(b)}])$. So x is the binary expansion of an absolutely normal number. □

6. Open problem

Many questions arise naturally from this work, but the following problem appears to be especially likely to demand new and useful methods.

As we have seen, normal numbers are closely connected to the theory of finite automata. Schnorr and Stimm [28] proved that normality is exactly the finite-state case of randomness. That is, a real number α is normal in a base $b \ge 2$ if and only if no finite-state automaton can make unbounded money betting on the successive digits of the base-b expansion of α with fair payoffs. The theory of finite-state dimension [11], which constrains Hausdorff dimension [17] to finite-state automata, assigns each real number α a finite-state dimension $\dim_{FS}^{(b)}(\alpha) \in [0, 1]$ in each base b. A real number α then turns out to be normal in base b if and only if $\dim_{FS}^{(b)}(\alpha) = 1$ [8]. Do there exist *absolutely dimensioned numbers*, i.e., real numbers α for which $\dim_{FS}(\alpha) = \dim_{FS}^{(b)}(\alpha)$ does not depend on b, and $0 < \dim_{FS}(\alpha) < 1$?

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The first author's research was supported in part by National Science Foundation Grants 0652569, 1143830, 1247051, 1545028, and 1900716. Part of this author's work was done during a sabbatical at Caltech and the Isaac Newton Institute for Mathematical Sciences at the University of Cambridge, part was done during three weeks at Heidelberg University, with support from the Mathematics Center Heidelberg and the Heidelberg University Institute of Computer Science, and part was done during the workshop "Normal Numbers: Arithmetical, Computational and Probabilistic Aspects" at the Erwin Schrödinger International Institute for Mathematics and Physics at the University of Vienna.

The second author's research was supported in part by Spanish Government MEC Grants TIN2011-27479-C04-01, TIN2016-80347-R, and PID2019-104358RB-I00. Part of this author's work was done during a research stay at the Isaac Newton Institute for Mathematical Sciences at the University of Cambridge, part was done during three weeks at Heidelberg University, with support from the Mathematics Center Heidelberg and the Heidelberg University Institute of Computer Science, and part was done during the workshop "Normal Numbers: Arithmetical, Computational and Probabilistic Aspects" at the Erwin Schrödinger International Institute for Mathematics and Physics at the University of Vienna.

We also thank two anonymous reviewers of this paper for very useful and detailed comments.

References

[1] C. Aistleitner, V. Becher, A.-M. Scheerer, T. Slaman, On the construction of absolutely normal numbers, Acta Arith. 180 (4) (2017) 333–346.
[2] D. Angluin, L. Valiant, Fast probabilistic algorithm for Hamiltonian circuits and matchings, J. Comput. Syst. Sci. 18 (1979) 155–193.
[3] V. Becher, S. Figueira, R. Picchi, Turing's unpublished algorithm for normal numbers, Theor. Comput. Sci. 377 (2007) 126–138.
[4] V. Becher, P.A. Heiber, T.A. Slaman, A polynomial-time algorithm for computing absolutely normal numbers, Inf. Comput. 232 (2013) 1–9.
[5] P. Billingsley, Probability and Measure, third edition, John Wiley and Sons, New York, N.Y., 1995.
[6] E. Borel, Sur les probabilités dénombrables et leurs applications arithmétiques, Rend. Circ. Mat. Palermo 27 (1) (1909) 247–271.
[7] J. Borwein, D. Bailey, Mathematics by Experiment: Plausible Reasoning in the 21st Century, second edition, A.K. Peters, 2008.
[8] C. Bourke, J.M. Hitchcock, N.V. Vinodchandran, Entropy rates and finite-state dimension, Theor. Comput. Sci. 349 (3) (2005) 392–406.
[9] Y. Bugeaud, Distribution Modulo One and Diophantine Approximation, vol. 193, Cambridge University Press, 2012.
[10] J.W.S. Cassels, On a problem of Steinhaus about normal numbers, Colloq. Math. 7 (1959) 95–101.
[11] J.J. Dai, J.I. Lathrop, J.H. Lutz, E. Mayordomo, Finite-state dimension, Theor. Comput. Sci. 310 (2004) 1–33.
[12] J. Doob, Regularity properties of certain families of chance variables, Trans. Am. Math. Soc. 47 (1940) 455–486.
[13] M. Feder, Gambling using a finite state machine, IEEE Trans. Inf. Theory 37 (1991) 1459–1461.
[14] S. Figueira, A. Nies, Feasible analysis and randomness, Manuscript, 2013.
[15] S. Figueira, A. Nies, Feasible analysis, randomness, and base invariance, Theory Comput. Syst. 56 (2015) 439–464.
[16] Y. Gurevich, S. Shelah, Nearly linear time, in: Proceedings of the First Symposium on Logical Foundations of Computer Science, Springer, 1989.
[17] F. Hausdorff, Dimension und äußeres Maß, Math. Ann. 79 (1919) 157–179.
[18] J. Hitchcock, J. Lutz, Why computational complexity requires stricter martingales, Theory Comput. Syst. 39 (2006) 277–296.
[19] D.E. Knuth, Construction of a random sequence, BIT Numer. Math. 5 (1965) 246–250.
[20] A.N. Kolmogorov, Foundations of the Theory of Probability, Chelsea, 1950.
[21] H. Lebesgue, Sur certaines demonstrations d'existence, Bull. Soc. Math. Fr. 45 (1917) 132–144.
[22] A. Lempel, J. Ziv, Compression of individual sequences via variable rate coding, IEEE Trans. Inf. Theory 24 (1978) 530–536.
[23] P. Lévy, Propriétés asymptotiques des sommes de variables indépendantes ou enchainées, J. Math. Pures Appl., Ser. 9 14 (4) (1935) 347–402.
[24] P. Lévy, Théorie de l'Addition des Variables Aleatoires, Gauthier-Villars, 1937 (second edition 1954).
[25] M. Madritsch, A. Scheerer, R. Tichy, Computable absolutely Pisot normal numbers, Acta Arith. 184 (2018) 7–29.
[26] E. Mayordomo, Construction of an absolutely normal real number in polynomial time, Manuscript, 2013.
[27] W.M. Schmidt, On normal numbers, Pac. J. Math. 10 (2) (1960) 661–672.
[28] C.-P. Schnorr, H. Stimm, Endliche automaten und zufallsfolgen, Acta Inform. 1 (4) (1972) 345–359.
[29] W. Sierpinski, Démonstration élémentaire du théorème de M. Borel sur les nombres absolument normaux et détermination effective d'une tel nombre, Bull. Soc. Math. Fr. 45 (1917) 125–132.
[30] A.M. Turing, On computable numbers with an application to the Entscheidungsproblem, Proc. Lond. Math. Soc. 42 (1936–1937) 230–265.
[31] A.M. Turing, A note on normal numbers, in: S.B. Cooper, J. van Leeuwen (Eds.), Alan Turing: His Work and Impact, Elsevier Science, 2013.
[32] J. Ville, Étude Critique de la Notion de Collectif, Gauthier–Villars, Paris, 1939.
[33] S. Wagon, Is π normal?, Math. Intell. 7 (1985) 65–67.