Automatic Kolmogorov complexity, normality and finite state dimension revisited

Alexander Kozachinskiy, Alexander Shen

Automatic Kolmogorov complexity, normality, and finite state dimension revisited

Alexander Kozachinskiy* Alexander Shen†

September 17, 2019

Abstract

It is well known that normality (all factors of a given length appear in an infinite sequence with the same frequency) can be described as incompressibility via finite automata. Still the statement and the proof of this result as given by Becher and Heiber [4] in terms of "lossless finite-state compressors" do not follow the standard scheme of Kolmogorov complexity definition (an automaton is used for compression, not decompression). We modify this approach to make it more similar to the traditional Kolmogorov complexity theory (and simpler) by explicitly defining the notion of automatic Kolmogorov complexity and using its simple properties. Other known notions (Shallit–Wang [39], Calude – Salomaa – Roblot [11]) of description complexity related to finite automata are discussed (see the last section).

Using this characterization, we provide easy proofs for most of the classical results about normal sequences, including the equivalence between aligned and non-aligned definitions of normality, the Piatetski-Shapiro sufficient condition for normality (in a strong form), and Wall's theorem saying that a normal real number remains normal when multiplied by a rational number or when a rational number is added. Using the same characterization, we prove a sufficient condition for normality of a sequence in terms of Kolmogorov complexity. This condition implies the normality of Champernowne's sequence as well as some generalizations of this result (provided by Champernowne himself, Besicovitch, Copeland and Erdös). It can be also used to give a simple proof of the result of Calude – Staiger – Stephan [12] saying that normality cannot be characterized in terms of the automatic complexity notion introduced by Calude – Salomaa – Roblot [11].

Then we extend this approach to finite state dimension showing that automatic Kolmogorov complexity can be used to characterize the finite state dimension (defined by Dai, Lathrop, Lutz and Mayordomo in [16]). We start with the block entropy definition of the finite state dimension given by Bourke, Hitchcock and Vinogradchandran [9] and show that one may use non-aligned blocks in this definition. Then we show that this definition is equivalent to the definition in terms

*HSE Computer Science Department, Moscow

†LIRMM, CNRS and University of Montpellier, France. On leave from IITP RAS, Moscow.
Supported by ANR-15-CE40-0016 RaCAF grant. Part of the work was done while visiting National Research University High School of Economics, Moscow.
E-mail: sasha.shen@gmail.com or alexander.shen@lirmm.fr

of automatic complexity. Finally, we use a slightly different version of automatic complexity (a finite state version of a priori complexity) to show the equivalence between the block entropy definition and original definition from [16] (this equivalence was proven in [9]). We also give a "machine-independent" characterization of finite state dimension in terms of superadditive functions that are "calibrated" in some sense (have not too many small values), or superadditive upper bounds for Kolmogorov complexity.

Finally, we use our tools to give a simple proof of Agafonov's result saying that normality is preserved by automatic selection rules [1] as well as the results of Schnorr and Stimm [38] that relate normality to finite state martingales.

Some results of this paper were presented at the Fundamentals in Computing Theory conferences in 2017 and 2019 [43, 21]. Preliminary version of this paper (that does not mention the finite state dimension) was published in `arxiv.org` in 2017 [42].

Contents

1 Introduction **4**

2 Automatic Kolmogorov complexity **6**
2.1 General scheme of defining complexities 6
2.2 Automatic description modes 7
2.3 Properties of automatic description modes 8
2.4 Properties of automatic complexity 9

3 Normality and incompressibility **12**
3.1 Normal sequences and numbers 12
3.2 Normality and incompressibility 13

4 Using the incompressibility criterion for normality **15**
4.1 Non-aligned normal sequences 15
4.2 Piatetski-Shapiro theorem 16
4.3 Sufficient condition for normality in terms of complexity 18
4.4 Copeland – Erdös' theorem 19
4.5 Besicovitch's theorem 20
4.6 Calude – Salomaa – Roblot question answered by Calude – Staiger – Stephan 21

5 Finite state dimension and automatic complexity **23**
5.1 Entropy rates for aligned and non-aligned blocks 23
5.2 Why do we need large blocks: a counterexample 27
5.3 Finite state dimension and Wall's theorem 29
5.4 Finite state dimension and automatic complexity 30
5.5 Machine-independent characterization of normal sequences and finite state dimension 36
5.6 Finite state martingales and automatic a priori complexity 38
5.7 Agafonov's and Schnorr–Stimm's theorems 43
5.8 Multi-account gales and a more general notion of finite state measures . . . 45

6 Discussion **46**

1 Introduction

What is an individual random object? When could we believe, looking at an infinite sequence α of zeros and ones, that α was obtained by tossing a fair coin? The minimal requirement is that zeros and ones appear "equally often" in α: both have limit frequency $1/2$. Moreover, it is natural to require that all 2^k bit blocks of length k appear equally often. Sequences that have this property are called *normal* (see the exact definition in Section 3.1; a historic account can be found in [4, 10]).

Intuitively, a reasonable definition of an individual random sequence should require much more than just normality; the corresponding notions are studied in the algorithmic randomness theory (see [18, 28] for the detailed exposition, [41] for a textbook and [40] for a short survey). The most popular randomness notion is called *Martin-Löf randomness*; the classical Schnorr – Levin theorem says that this notion is equivalent to *incompressibility*: a sequence α is Martin-Löf random if an only if prefixes of α are incompressible (do not have short descriptions). See again [18, 28, 41, 40] for exact definitions and proofs.

It is natural to expect that normality, being a weak randomness property, corresponds to some weak incompressibility property. The connection between normality and finite-state computations was noticed long ago, as the title of [1] shows. This connection led to a characterization of normality as "finite-state incompressibility" (see [4]). However, the notion of incompressibility that was used in [4] does not fit well the general framework of Kolmogorov complexity (finite automata are considered there as *compressors*, while in the usual definition of Kolmogorov complexity we restrict the class of allowed *decompressors*).

In this paper we give a definition of automatic Kolmogorov complexity that restricts the class of allowed decompressors and is suitable for the characterization of normal sequences as incompressible ones. This definition and its properties are considered in Section 2. Section 3 presents one of our main results: characterization of normality in terms of automatic complexity. First (Section 3.1) we recall the notion of a normal sequence. Then (Section 3.2) we provide a characterization of normal sequences as sequence whose prefixes have automatic Kolmogorov complexity close to the length.

This characterization is used in Section 4 to provide simple proofs for many classical results about normal sequences. In Section 4.1 we give a simple proof of an old result (Borel, Pillai, Niven – Zuckerman, [8, 34, 30]) saying that normality can be equivalently defined in terms of frequencies of aligned or non-aligned blocks (we get the same notion in both cases). In Section 4.2 we provide a simple proof of the result by Piatetski-Shapiro [31, 32] saying that a sequence is normal if for every k-bit block its frequency is not much bigger than its expected frequency in a random sequence. This proof can be used to prove a stronger version of this result, replacing a constant factor in Piatetski-Shapiro version by factor $2^{o(k)}$. We note also that Piatetski-Shapiro's result easily implies Wall's theorem (saying that normal numbers remain normal when multiplied by a rational factor).

Then in Section 4.3 we return to the first example of a normal sequence given by

Champernowne [14] and show that its normality easily follows from a simple sufficient condition for normality in terms of Kolmogorov complexity (Theorem 5) . The same sufficient condition easily implies the generalizations of Champernowne's results obtained by Copeland–Erdös ([15], see Section 4.4) and Besicovitch ([6], see Section 4.5). Finally, in Section 4.6 we show how this sufficient condition gives a simple proof of a result proven by Calude, Staiger and Stephan [12]. It says that the definition of automatic complexity from [11] does not provide a criterion of normality (this question was asked in [11]).

The notion of normality can be interpreted as "weak randomness" (weak incompressibility). Instead of randomness, one can consider a more general notion of *effective Hausdorff dimension* introduced by Lutz in [24] (see [41, Sections 5.8 and 9.10] for details). The effective Hausdorff dimension is defined for arbitrary binary sequences and is between 0 and 1; the smaller it is, the more compressible is the sequence. Formally speaking, the effective Hausdorff dimension of a sequence $\alpha = a_0a_1\ldots$ can be defined in terms of Kolmogorov complexity as $\liminf_n[(\text{complexity of } a_0\ldots a_{n-1})/n]$. For random sequences the effective Hausdorff dimension equals 1 (as well as for some non-random sequences, e.g., for a sequence that is obtained from a random one by replacing all terms a_{2^n} by zeros). This notion is an effective counterpart of the classical notion of Hausdorff dimension, see [24, 41].

The notion of effective Hausdorff dimension has a scaled-down version where we restrict ourselves by finite automata. This parallel notion is called *finite state dimension* and was introduced in [16]. In this paper it was defined in terms of finite-state martingales; in [9] an equivalent definition in terms of entropy rates was provided. In Section 5.1 we revisit the definition of finite state dimension in terms of entropy rates and show that one may use both aligned and non-aligned blocks in this definition and get the same quantity. However, this equivalence works does not work for blocks of fixed size, as the counterexamples of Section 5.2 (Theorem 9) show. In Section 5.3 we give a simplified proof of a theorem of Doty, Lutz and Nandakumar [17] saying that finite state dimension does not change when a real number is multiplied by a rational factor. Then in Section 5.4 we show that finite state dimension can be characterized in terms of automatic complexity as the lim inf of complexity/length ratio for prefixes, thus giving a characterization of finite state dimension that is parallel to the characterization of effective Hausdorff dimension in terms of Kolmogorov complexity. The only difference with effective Hausdorff dimension is that we use automatic Kolmogorov complexity instead of the standard one (and have to take infimum over all automata since there is no universal automaton that leads to minimal complexity). To prove this characterization, we use the definition of finite-state dimension in terms of entropy rates. In Section 5.5 we show that this characterization is quite robust: automatic complexity can be replaced by other similar notions. For example, we may consider all superadditive upper bounds for Kolmogorov complexity (Theorem 12), or even give a characterization of finite-state dimension (Theorem 13) that does not mention entropy, Kolmogorov complexity and finite-state machines at all, and just considers a class of superadditive functions that are "calibrated" in some natural sense (have not too many small values).

In Section 5.6 we give a simple proof that the definition of finite state dimension in terms of entropy rates is equivalent to the original definition from [16]. For that we discuss a finite-state version of the notion of a priori probability (maximal semimeasure) used in the algorithmic information theory, and show that it also can be used to characterize finite state dimension. In Section 5.7 we use our tools to give a simple proof for the result of Agafonov [1] (finite automaton selects a normal sequence if applied to a normal sequence) and its extension by Schnorr and Stimm [38] saying the any finite-state martingale is either constant or exponentially decreases on sufficiently long prefixes of a normal sequence. We also mention a natural notion of finite-state measure that generalizes the notion of multi-account gales [16] and also can be used to characterize finite-state dimension (Section 5.8).

Finally, in Section 6 we compare our definition of automatic complexity with other similar notions.

2 Automatic Kolmogorov complexity

2.1 General scheme of defining complexities

Let us recall the definition of algorithmic (Kolmogorov) complexity. It is usually defined in the following way: $\mathrm{C}(x)$, the complexity of an object x, is the minimal length of its "description". We assume that both objects and descriptions are binary strings; the set of binary strings is denoted by $\mathbb{B}^*$, where $\mathbb{B} = \{0, 1\}$. Of course, this definition makes sense only after we explain which type of "descriptions" we consider, but most versions of Kolmogorov complexity can be defined according to this scheme [44].

Definition 1. Let $D \subset \mathbb{B}^* \times \mathbb{B}^*$ be a binary relation; we read $(p, x) \in D$ as "p is a D-description of x". Then *complexity function* C_D is defined as

$$\mathrm{C}_D(x) = \min\{|p| \colon (p, x) \in D\},$$

i.e., as the minimal length of a D-description of x.

Here $|p|$ stands for the length of a binary string p and $\min(\varnothing) = +\infty$, as usual. We say that D is a *description mode* and $\mathrm{C}_D(x)$ is the *complexity of x with respect to the description mode D*.

We get the original version of Kolmogorov complexity ("plain complexity") if we consider all computable functions as description modes, i.e., if we consider relations $D_f = \{(p, f(p))\}$ for arbitrary computable partial functions f as description modes. Equivalently, we may say that we consider (computably) enumerable relations D that are graphs of functions (for every p there exists at most one x such that $(p, x) \in D$; each description describes at most one object). Then the Kolmogorov – Solomonoff optimality theorem says that there exists an optimal D in this class that makes C_D minimal up to an $O(1)$ additive term. We assume that the reader is familiar with basic properties of Kolmogorov complexity, see, e.g., [23, 41]; for a short introduction see also [40].

Note that we could get a trivial C_D if we take, e.g., the set of all pairs as a description mode D. In this case all strings have complexity zero, since the empty string describes all of them. So we should be careful and do not consider description modes where the same string describes too many different objects.

2.2 Automatic description modes

To define our class of description modes, let us first recall some basic notions related to finite automata. Let A and B be two finite alphabets. Consider a directed graph G whose edges are labeled by pairs (a, b) of letters (from A and B respectively). We also allow pairs of the form (a, ε), (ε, b), and $(\varepsilon, \varepsilon)$ where ε is a special symbol (not in A or B) that informally means "no letter". For such a graph, consider all directed paths in it (no restriction on starting or final points), and for each path concatenate all the first components and also (separately) all the second components of the pairs along the path; ε is replaced by an empty word. For each path we get some pair (u, v) where $u \in A^*$ and $v \in B^*$ (i.e., u and v are words over alphabets A and B). Consider all pairs that can be read in this way along all paths in G. For each labeled graph G we obtain a relation (set of pairs) R_G that is a subset of $A^* \times B^*$. For the purposes of this paper, we call the relations obtained in this way "automatic". This notion is similar to *rational relations* defined by transducers [5, Section III.6]. The difference is that we do not fix initial/finite states (so every sub-path of a valid path is also valid) and that we do not allow arbitrary words as labels, only letters and ε. (This will be important, e.g., for the statement (j) of Theorem 1.)

Definition 2. A relation $R \subset A^* \times B^*$ is *automatic* if there exists a labeled graph (*automaton*) G such that $R = R_G$.

Now we define automatic description modes as automatic relations where each string describes at most $O(1)$ objects.

Definition 3. A relation $D \subset \mathbb{B}^* \times \mathbb{B}^*$ is an *automatic description mode* if

- D is automatic in the sense of Definition 2;
- D is a graph of an $O(1)$-valued function: there exists some constant c such that for each p there are at most c values of x such that $(p, x) \in D$.

For every automatic description mode D we consider the corresponding complexity function C_D. There is no optimal mode D that makes C_D minimal (see Theorem 1 below). So, stating some properties of complexity, we need to mention D explicitly. Moreover, for a statement that compares the complexities of different strings, we need to say something like "for every automatic description mode D there exists another automatic description mode D' such that…", and then make a statement that involves both C_D and $\mathrm{C}_{D'}$. (A similar approach is needed when we try to adapt inequalities for Kolmogorov complexity to the case of resource-bounded complexities.)

2.3 Properties of automatic description modes

Let us first mention some basic properties of automatic description modes.

Proposition 1.

(a) *The union of two automatic description modes is an automatic description mode.*

(b) *The composition of two automatic description modes is an automatic description mode.*

(c) *If D is a description mode, then $\{(p, x0)\colon (p, x) \in D\}$ is a description mode (here $x0$ is the binary string x with 0 appended); the same is true for $x1$ instead of $x0$.*

Proof. There are two requirements for an automatic description mode: (1) the relation is automatic and (2) the number of images is bounded. The second one is obvious in all three cases. The first one can be proven by a standard argument (see, e.g., [5, Theorem 4.4]) that we reproduce for completeness.

(a) The union of two relations R_G and R'_G for two automata G and G' corresponds to an automaton that is a disjoint union of G and G'.

(b) Let S and T be automatic relations that correspond to automata K and L. Consider a new graph that has set of vertices $K \times L$. (Here we denote an automaton and the set of vertices of its underlying graph by the same letter.)

- If an edge $k \to k'$ with a label (a, ε) exists in K, then the new graph has edges $(k, l) \to (k', l)$ for all $l \in L$; all these edges have the same label (a, ε).
- In the same way an edge $l \to l'$ with a label (ε, c) in L causes edges $(k, l) \to (k, l')$ in the new graph for all k; all these edges have the same label (ε, c).
- Finally, if K has an edge $k \to k'$ labeled (a, b) and at the same time L has an edge $l \to l'$ labeled (b, c), where b is the same letter, then we add an edge $(k, l) \to (k', l')$ labeled (a, c) in the new graph.

Any path in the new graph is projected into two paths in K and L. Let (p, q) and (u, v) be the pairs of words that can be read along these projected paths in K and L respectively, so $(p, q) \in S$ and $(u, v) \in T$. The construction of the graph $K \times L$ guarantees that $q = u$ and that we read (p, v) in the new graph along the path. So every pair (p, v) of strings that can be read in the new graph belongs to the composition of S and T.

On the other hand, assume that (p, v) belong to the composition, i.e., there exists q such that (p, q) can be read along some path in K and (q, v) can be read along some path in L. Then the same word q appears in the second components in the first path and in the first components in the second path. If we align the two paths in such a way that the letters of q appear at the same time, we get a valid transition of the third type for each letter of q. Then we complete the path by adding transitions in between the synchronized ones (interleaving them in arbitrary way); all these transitions exist in the new graph by construction.

(c) We add an additional outgoing edge labeled $(\varepsilon, 0)$ for each vertex of the graph; all these edges go to a special vertex that has no outgoing edges. $\square$

Remark. Given a graph, one can check in polynomial time whether the corresponding relation is $O(1)$-valued [47, Theorem 5.3, p. 777].

2.4 Properties of automatic complexity

Now we are ready to prove the following simple result about the properties of *automatic Kolmogorov complexity* functions, i.e., of functions C_R where R is some automatic description mode.

Theorem 1 (Basic properties of automatic Kolmogorov complexity)**.**

(a) *There exists an automatic description mode R such that $\mathrm{C}_R(x) \leqslant |x|$ for all strings x.*

(b) *For every automatic description mode R there exists some automatic description mode R' such that $\mathrm{C}_{R'}(x0) \leqslant \mathrm{C}_R(x)$ and $\mathrm{C}_{R'}(x1) \leqslant \mathrm{C}_R(x)$ for all x.*

(c) *For every automatic description mode R there exists some automatic description mode R' such that $\mathrm{C}_{R'}(\bar{x}) \leqslant \mathrm{C}_R(x)$, where $\bar{x}$ stands for the reversed x.*

(d) *For every automatic description mode R there exists some constant c such that $\mathrm{C}(x) \leqslant \mathrm{C}_R(x) + c$. (Here C stands for the plain Kolmogorov complexity.)*

(e) *For every automatic description mode R there exists some constant c such that for every n there is at most $c2^n$ strings x such that $\mathrm{C}_R(x) < n$.*

(f) *For every $c > 0$ there exists an automatic description mode R such that $\mathrm{C}_R(1^n) \leqslant n/c$ for all n.*

(g) *For every automatic description mode R there exists some $c > 0$ such that $\mathrm{C}_R(1^n) \geqslant n/c - 1$ for all n.*

(h) *For every two automatic description modes R_1 and R_2 there exists an automatic description mode R such that $\mathrm{C}_R(x) \leqslant \mathrm{C}_{R_1}(x)$ and $\mathrm{C}_R(x) \leqslant \mathrm{C}_{R_2}(x)$ for all x.*

(i) *There is no optimal automatic description mode. (A mode R is called optimal in some class if for every mode R' in this class there exists some c such that $\mathrm{C}_R(x) \leqslant \mathrm{C}_{R'}(x) + c$ for all strings x.)*

(j) *For every automatic description mode R, if x' is a substring of x, then $\mathrm{C}_R(x') \leqslant \mathrm{C}_R(x)$.*

(k) *Moreover, $\mathrm{C}_R(xy) \geqslant \mathrm{C}_R(x) + \mathrm{C}_R(y)$ for every two strings x and y.*

(l) *For every automatic description mode R and for every constant $\varepsilon > 0$ there exists an automatic description mode R' such that $\mathrm{C}_{R'}(xy) \leqslant (1+\varepsilon)\,\mathrm{C}_R(x) + \mathrm{C}_R(y)$ for all strings x and y.*

(m) *Let S be an automatic description mode. Then for every automatic description mode R there exists an automatic description mode R' such that* $\mathrm{C}_{R'}(y) \leqslant \mathrm{C}_R(x)$ *for every* $(x, y) \in S$.

(n) *If we allow a bigger alphabet B instead of* $\mathbb{B} = \{0, 1\}$ *as an alphabet for descriptions, we divide the complexity by* $\log |B|$, *up to a constant factor that can be chosen arbitrarily close to* 1.

Proof. (a) Consider an identity relation as a description mode; it corresponds to an automaton with one state.

(b) This is a direct corollary of Proposition 1, (c).

(c) The definition of an automaton is symmetric (all edges can be reversed), and the $O(1)$-condition still holds.

(d) Let R be an automatic description mode. An automaton defines a decidable (computable) relation, so R is decidable. Since R defines a $O(1)$-valued function, a Kolmogorov description of some y that consists of its R-description x and the ordinal number of y among all strings that are in R-relation to x (in some natural ordering), is only $O(1)$ bits longer than x.

(e) This is a direct corollary of d, since there is less than 2^n strings of Kolmogorov complexity less than n. Or we may just count all the descriptions of length less than n. There is less than 2^n of them, and each describes only $O(1)$ strings.

(f) Consider an automaton that consists of a cycle where it reads one input symbol 1 and then produces c output symbols 1. (Since we consider the relation as an $O(1)$-multivalued function, we sometimes consider the first components of pairs as "input symbols" and the second components as "output symbols".) Recall that there is no restrictions on initial and finite states, so this automaton produces all pairs $(1^k, 1^l)$ where $(k-1)c \leqslant l \leqslant (k+1)c$.

(g) Consider an arbitrary description mode, i.e., an automaton that defines some $O(1)$-valued relation. Then every cycle in the automaton that produces some output letter should also produce some input letter, otherwise an empty input string corresponds to infinitely many output strings. For any sufficiently long path in the graph we can cut away a minimal cycle, removing at least one input letter and at most c output letters, where c is the number of states, until we get a path of length less than c.

(h) This follows from Proposition 1, (a).

(i) This statement is a direct consequence of (f) and (g). Note that for finitely many automatic description modes there is a mode that is better than all of them, as (h) shows, but we cannot do the same for all description modes (as was the case for Kolmogorov complexity).

(j) If R is a description mode, (p, x) belongs to R and x' is a substring of x, then there exists some substring p' of p such that $(p', x') \in R$. Indeed, we may consider the input symbols used while producing x'.

(k) Note that in the previous argument we can choose disjoint p' for disjoint x'.

(l) Informally, we modify the description mode as follows: a fixed fraction of input symbols is used to indicate when a description of x ends and a description of y begins.

More formally, let R be an automatic description mode; we use the same notation R for the corresponding automaton. Consider $N+1$ copies of R (called 0-, 1-,…, N-th layers). The outgoing edges from the vertices of i-th layer that contain an input symbol are redirected to $(i+1)$-th layer (the new state remains the same, only the layer changes, so the layer number counts the input length). The edges with no input symbol are left unchanged (and go to i-th layer as before). The edges from the N-th layer are of two types: for each vertex x there is an edge with label $(0, \varepsilon)$ that goes to the same vertex in 0-th layer, and edges with labels $(1, \varepsilon)$ that connect each vertex of N-th layer to all vertices of an additional copy of R (so we have $N+2$ copies in total). If both x and y can be read (as outputs) along the edges of R, then xy can be read, too (additional zeros should be added to the input string after groups of N input symbols). We switch from x to y using the edge that goes from Nth layer to the additional copy of R (using additional symbol 1 in the input string). The overhead in the description is one symbol per every N input symbols used to describe x. We get the required bound, since N can be arbitrarily large.

The only thing to check is that the new automaton is $O(1)$-valued. Indeed, the possible switch position (when we move to the states of the additional copy of R) is determined by the positions of the auxiliary bits modulo $N+1$: when this position modulo $N+1$ is fixed, we look for the first 1 among the auxiliary bits. This gives only a bounded factor $(N+1)$ for the number of possible outputs that correspond to a given input.

(m) The composition $S \circ R$ is an automatic description mode due to Proposition 1, (b).

(n) Take the composition of a given description mode R with a mode that provides block encoding of inputs. Note that block encoding can be implemented by an automaton. There is some overhead when $|B|$ is not a power of 2, but the corresponding factor becomes arbitrarily close to 1 if we use block code with large block size. □

Remark. Not all these results are used in the sequel; we provide them for comparison with the properties of the standard Kolmogorov complexity function. The bottom line is that *automatic complexity is an upper bound for Kolmogorov complexity that has the "superadditivity" property* $\mathrm{C}_R(xy) \geqslant \mathrm{C}_R(x) + \mathrm{C}_R(y)$, see items (d) and (k). Note that usual Kolmogorov complexity is not superadditive for obvious reasons: say, $\mathrm{C}(xx)$ is close to $\mathrm{C}(x)$, not to $2\,\mathrm{C}(x)$.

As we will see (Sections 5.4 and 5.5), the characterization of normality (and finite state dimension, see below) in terms of automatic complexity can be extended to all upper bounds for Kolmogorov complexity that are superadditive, and also to all superadditive functions satisfying the property (e).

3 Normality and incompressibility

3.1 Normal sequences and numbers

Consider an infinite bit sequence $\alpha = a_0a_1a_2\ldots$ and some integer $k \geqslant 1$. Split the sequence α into k-bit blocks: $\alpha = A_0A_1\ldots$. For every k-bit string r consider the limit frequency of r among the A_i, i.e. the limit of $\#\{i\colon i < N \text{ and } A_i = r\}/N$ as $N \to \infty$. This limit may exist or not; if it exists for some k and for all r, we get a probability distribution on k-bit strings.

Definition 4. A sequence α is *normal* if for every number k and every string r of length k this limit exists and is equal to 2^{-k}.

Sometimes sequences with these properties are called *strongly normal* while the name "normal" is reserved for sequences that have this property for $k = 1$.

There is a version of the definition of normal sequences that considers *all* occurrences of some string r in α (while Definition 4 considers only *aligned* ones, whose starting point is a multiple of k). In this "non-aligned" version we require that the limit of $\#\{i < N : \alpha_i\alpha_{i+1}\ldots\alpha_{i+k-1} = r\}/N$ equals 2^{-k} for all k and for all strings r of length k. A classical result[1] says that this is an equivalent notion, and we give below (Section 4.1) a simple proof of this equivalence using automatic complexity. Before this proof is given, we will distinguish the two definitions by using the name "non-aligned-normal" for the second version.

A real number is called *normal* if its binary expansion is normal (we ignore the integer part). If a number has two binary expansions, like $0.0111\ldots = 0.1000\ldots$, both expansions are not normal, so this is not a problem.

A classical example of a normal number is the *Champernowne number* [14]

$$0.0\,1\,10\,11\,100\,101\,110\,111\,1000\,1001\ldots$$

(the concatenation of all positive integers in binary). Let us sketch the proof of its normality (not used in the sequel) using the non-aligned version of normality definition. All N-bit numbers in the Champernowne sequence form a block that starts with 10^{N-1} and ends with 1^N. Note that every string of length $k \ll N$ appears in this block with

[1] In fact, this result has a rather complicated history. The original definition of normal numbers was given by Borel [8]. He required that every k-bit strings appears with frequency 2^{-k} among blocks that we get when we delete some finite prefix of the sequence and cut the rest into k-bit blocks. This implies both aligned and non-aligned normality (the aligned normality is the special case when the prefix is empty, the non-aligned normality can be shown by averaging frequencies for prefixes of length $0, 1, \ldots, k-1$). Borel noted that his definition follows from non-aligned normality ("La propriété caractéristique", p. 261). However, he gave no proof, and the relation between these three definitions (aligned, non-aligned and Borel's definition that implies both) was clarified much later. Pillai [34], correcting his earlier paper [33], showed that aligned normality implies Borel's definition. Niven and Zuckerman [30] gave a proof of Borel's claim. Cassels [13] provided an alternative proof for the result of Niven and Zuckerman, while Maxfield [27] provided an alternative proof for the result of Pillai. See also [29]; a more recent exposition can be found, e.g., [22, Chapter 1, Section 8]; it uses as a tool the Piatetski-Shapiro criterion (see Section 4.2 below).

probability close to 2^{-k}, since each of 2^{N-1} strings (after the leading 1 for the N-bit numbers in the Champernowne sequence) appears exactly once. The deviation is caused by the leading 1's and also by the boundaries between the consecutive N-bit numbers where the k-bit substrings are out of control. Still the deviation is small since $k \ll N$.

This is not enough to conclude that C is (non-aligned) normal, since the definition speaks about frequencies in all prefixes; the prefixes that end on a boundary between two blocks are not enough. The problem appears because the size of a block is comparable to the length of the prefix before it. To deal with arbitrary prefixes, let us note that if we ignore *two* leading digits in each number (first 10 and then 11) instead of one, the rest is periodic in the block (the block consists of two periods). If we ignore three leading digits, the block consists of four periods, etc. An arbitrary prefix is then close to the boundary between these sub-blocks, and the distance can be made small compared to the total length of the prefix. (End of the proof sketch.)

In fact, the full proof that follows this sketch is quite tedious. There are much more general reasons why this number is normal, as we will see in Section 4.3, where this result becomes an immediate corollary of the sufficient condition for normality in terms of Kolmogorov complexity (and this condition in its turn is a easy consequence of the criterion of normality in terms of automatic complexity).

The definition of normality can be given for an arbitrary alphabet (instead of the binary one), and we get the notion of *b-normality* of a real number for every base $b \geqslant 2$. It is known that for different bases we get non-equivalent notions of normal real numbers (a rather difficult result). The numbers in $[0, 1]$ that are normal for every base are called *absolutely normal*. Their existence can be proved by a probabilistic argument. Indeed, for every base b, almost all reals are b-normal (the non-normal numbers have Lebesgue measure 0 by the Strong Law of Large Numbers). Therefore the numbers that are not absolutely normal form a null set (a countable union of the null sets for each b). The constructive version of this argument shows that there exist computable absolutely normal numbers. This result goes back to an unpublished note of Turing (1938, see [2]).

In the next section we prove the connection between normality and automatic complexity: a sequence α is normal if for every automatic description mode D the corresponding complexity C_D of its prefix never becomes much smaller than the length of that prefix.

3.2 Normality and incompressibility

Theorem 2. *A sequence $\alpha = a_0a_1a_2\ldots$ is normal if and only if*

$$\liminf_{n\to\infty} \frac{\mathrm{C}_R(a_0a_1\ldots a_{n-1})}{n} \geqslant 1$$

for every automatic description mode R.

Proof. First, let us show that a sequence that is not normal is compressible. Assume that for some bit sequence α and for some k the requirement for aligned k-bit blocks is not satisfied. Using compactness arguments, we can find a sequence of lengths N_i such that for the prefixes of these lengths the frequencies of k-bit blocks do converge to some probability distribution A on $\mathbb{B}^k$, but this distribution is not uniform. Then its Shannon entropy $H(A)$ is less than k.

The Shannon theorem can then be used to construct a block code of average length close to $H(A)$, namely, of length at most $H(A)+1$ (this "+1" overhead is due to rounding if the frequencies are not powers of 2). Since this code can be easily converted into an automatic description mode, it will give the desired result if $H(A) < k-1$. It remains to show that it is the case for long enough blocks.

Selecting a subsequence, we may assume without loss of generality that the limit frequencies exist also for (aligned) $2k$-bit blocks, so we get a random variable A_0A_1 whose values are $2k$-bit blocks (and A_0 and A_1 are their first and second halves of length k). The variables A_0 and A_1 may be dependent, and their distributions may differ from the initial distribution A for k-bit blocks. Still we know that A is the average of A_0 and A_1 (since A is computed for all blocks, and A_0 [resp. A_1] corresponds to odd [resp. even] blocks). A convexity argument[2] (the function $p \mapsto -p \log p$ used in the definition of entropy has negative second derivative) shows that $H(A) \geqslant [H(A_0) + H(A_1)]/2$. Then

$$H(A_0A_1) \leqslant H(A_0) + H(A_1) \leqslant 2H(A),$$

so A_0A_1 has twice bigger difference between entropy and length (at least). Repeating this argument, we can find k such that the difference between length and entropy is greater than 1. This finishes the proof in one direction.

Now we need to prove that an arbitrary normal sequence α is incompressible. Let R be an arbitrary automatic description mode. Consider some k and split the sequence into k-bit blocks $\alpha = A_0A_1A_2\ldots$. (Now A_i are just the blocks in α, not random variables). We will show that

$$\liminf \mathrm{C}_R(A_0A_1\ldots A_{n-1})/nk$$

cannot be much smaller than 1. More precisely, we will show that

$$\liminf \frac{\mathrm{C}_R(A_0A_1\ldots A_{n-1})}{nk} \geqslant 1 - \frac{O(1)}{k},$$

where the constant in $O(1)$ does not depend on k. This is enough, because (i) adding the last incomplete block can only increase the complexity and the change in length is negligible, and (ii) the value of k may be arbitrarily large.

Now let us prove this bound for some fixed k. Recall that

$$\mathrm{C}_R(A_0A_1\ldots A_{n-1}) \geqslant \mathrm{C}_R(A_0) + \mathrm{C}_R(A_1) + \ldots + \mathrm{C}_R(A_{n-1})$$

and that $\mathrm{C}(x) \leqslant \mathrm{C}_R(x) + O(1)$ for all x and some $O(1)$-constant that depends only on R (Theorem 1). By assumption, all k-bit strings appear with the same limit frequency

[2]There is more general way to explain this: consider a random bit b and random variable A' that has the same distribution as A_0 when $b=0$ and the same distribution as A_1 when $b=1$. Then A' has the same distribution as A, and $H(A') = H(A)$ is not smaller than $H(A'|b) = [H(A_0) + H(A_1)]/2$.

among A_0, A_1,..., A_{n-1}. It remains to note that the average Kolmogorov complexity $\mathrm{C}(x)$ of all k-bit strings is $k - O(1)$; indeed, the fraction of k-bit strings that can be compressed by more than d bits ($\mathrm{C}(x) < k - d$) is at most 2^{-d}, and the series $\sum d2^{-d}$ (the upper bound for the average number of bits saved by compression) has finite sum.

Alternatively, one may also note that for $d = \log k$ we have $O(1/k)$ fraction of strings that are compressible more than by d (and at most by k) bits, and all other strings are compressible at most by $d = \log k$ bits, so the average compression is $O(1) + O(\log k) = O(\log k)$, and $O(\log k)$ bound in enough for our purposes (we do not need the stricter $O(1)$ bound proven earlier). □

4 Using the incompressibility criterion for normality

In this section we use the incompressibility characterization of normality to provide simple proofs for several classical results about normal sequences. First we prove that one may consider non-aligned frequencies of blocks when defining normality (Section 4.1). Then we give a simple proof of Piatetski-Shapiro's theorem from [31, 32] (Section 4.2). We give a simple sufficient condition for the normality of Champernowne-type sequences (Theorem 5, Section 4.3). This condition implies the normality of Champernowne's sequence; it is then applied to provide simple proofs of the results from Copeland–Erdös [15] (Section 4.4), Besicovitch [6] (Section 4.5) and Calude – Staiger – Stephan (Section 4.6).

4.1 Non-aligned normal sequences

Recall the proof of Theorem 2. A small modification of this proof adapts it to the non-aligned definition of normality, thus providing the proof of the equivalence between aligned and non-aligned definitions of normality. Let us see how this is done.

Let α be a sequence that is not normal in the non-aligned version. This means that for some k the (non-aligned) k-bit blocks do not have the correct limit distribution. These blocks can be split into k groups according to their starting positions modulo k. In one of the groups blocks do not have a correct limit distribution (otherwise the average distribution would be correct, too). So we can delete some prefix (less than k symbols) of our sequence and get a sequence that is not normal in the aligned sense. Its prefixes are compressible (as we have seen). The same is true for the original sequence since adding a fixed finite prefix (or suffix) changes complexity and length at most by $O(1)$, after a suitable change of the description mode, as Theorem 1, (a,b), implies.

In the other direction: let us assume that the sequence is normal in the non-aligned sense. The aligned frequency of some compressible-by-d-bits block (as well as any other block) can be only k times bigger than its non-aligned frequency, which is exponentially small in d (the number of saved bits), so we can choose the parameters to get the required bound.

Here are the details. Let us consider blocks (strings) of length k whose C_R-complexity is smaller than $k - d$. There is at most $O(2^{k-d})$ of them, as Theorem 1(e) says. So their frequency among *aligned* blocks in a *non-aligned normal* sequence is at most $k2^{-d+O(1)}$. For all other blocks R-compression saves at most d bits, and for d-compressible blocks it saves at most k bits, so the average number of saved bits (per k-bit block) is bounded by

$$d + k2^{-d+O(1)} \cdot k = d + O(k^2 2^{-d}).$$

We need this bound to be $o(k)$, i.e., we need that

$$\frac{d}{k} + O(k2^{-d}) = o(1)$$

as $k \to \infty$. This can be achieved, for example, if $d = 2\log k$.

In this way we get the following corollary [8, 34, 30]:

Corollary. *The aligned and non-aligned definitions of normality are equivalent.*

Note also that adding/deleting a finite prefix does not change the compressibility, and, therefore, normality. (For the non-aligned version of the normality definition it is obvious anyway, but for the aligned version it is not so easy to see directly, see the discussion of the original Borel's definition of normality above for historical details.)

4.2 Piatetski-Shapiro theorem

Piatetski-Shapiro in [31] proved[3] the following result: *if for some constant c and for all k every k-bit block appears in a sequence α with* (*non-aligned*) $\limsup$-*frequency at most $c2^{-k}$, then the sequence α is normal.*

This result is an immediate byproduct of the proof of the normality criterion (Theorem 2). Indeed, in the argument above we had a constant factor in the $O(k2^{-d})$ bound of Section 4.1 for the average compression due to compressible blocks. If the compressible blocks appear at most c times more often (as well as all other blocks, but this does not matter), we still have the same O-bound, so we get Piatetski-Shapiro's result (in aligned and non-aligned version at the same time; Piatetski-Shapiro considered the aligned version).

We can even allow the constant c to depend on k if its growth as a function of k is not too fast. Namely, the following stronger result was proven by Piatetski-Shapiro in [32]):

Theorem 3 (Piatetski-Shapiro theorem, strong version)**.** *Let α be an infinite bit sequence. Assume that for every k and for every k-bit block B its aligned* (*or non-aligned*) *frequency in all sufficiently long prefixes of α does not exceed $c_k 2^{-k}$, where c_k depends only on k and $c_k = 2^{o(k)}$. Then α is a normal sequence.*

Proof. Note first that the non-aligned version of this result follows for the aligned version. Indeed, aligned frequency of arbitrary block may exceed the non-aligned frequency of

[3] The proof used ergodic theory. Later in [35, 36, 32] alternative proofs that do not refer to ergodic theory were given.

the same block only by a factor of k, and $k = 2^{o(k)}$, so this additional factor still keeps $c_k = 2^{o(k)}$.

To prove the aligned version of the result, recall the proof of Theorem 2. Consider some threshold d_k (to be chosen later). We split all k-bit blocks we into two groups: the blocks that are compressible by more than d_k bits, and all the other ones. The fraction of the blocks of the first type (called "compressible" blocks in the sequel) among all k-bit strings is at most 2^{-d_k}. Therefore, by the assumption, if we split a long prefix of α into aligned k-bit blocks, the fraction of compressible blocks among them is bounded by (approximately) $c_k 2^{-d_k}$, and each of them is compressible by at most k bits (for obvious reasons). All other blocks are compressible by at most d_k bits, so the number of saved bits per block is at most $kc_k 2^{-d_k} + d_k$. We need this amount to be $o(k)$ to finish the proof of normality as before, so we need to choose d_k in such a way that

$$d_k = o(k) \quad \text{and} \quad kc_k 2^{-d_k} = o(k).$$

The second condition says that d_k exceeds $\log c_k$ and the difference tends to infinity. Since $c_k = 2^{o(k)}$, one can easily satisfy both conditions (e.g., let d_k be $\log c_k + \log k$). $\square$

Remark. In fact, Piatetski-Shapiro's statement in [32] is a bit stronger: he assumes only that $c_k = 2^{o(k)}$ for infinitely many k, i.e., that $\liminf_k \frac{\log c_k}{k} = 0$. The proof remains the same (we use the assumption to get a lower bound for automatic complexity of a prefix by splitting it into blocks; it is enough to do this not for all k but for infinitely many k). Also there is a minor technical difference: he considered real numbers α and the distribution of fractional parts of αq^k, where q is the base (we consider the case $q = 2$, but this does not matter). The condition in [32] says that the density of fractional parts that fall inside some interval $\Delta \subset [0, 1]$ is bounded by $f(|\Delta|)$ for a suitable f, where $|\Delta|$ is the length of Δ. It is easy to see that one can consider only intervals Δ obtained by dividing $[0, 1]$ into $q, q^2, \ldots$ parts (since any interval can be covered by these "aligned" intervals with bounded overhead). In this way we get the statement formulated above.[4]

Remark. The bound for c_k in this theorem is optimal, as the following example (from [32]) shows. Consider a sequence α that is random with respect to Bernoulli measure with parameter $\frac{1}{2} + \delta$. Then the frequency of the most frequent k-bit block (all ones) is $(\frac{1}{2} + \delta)^n = 2^{-n} 2^{\varepsilon n}$ for some constant ε that can be arbitrarily small if δ is small. On the other hand, α is not normal.

Let us note that Piatetski-Shapiro's result easily implies a result of Wall [46]. Recall that a real number is normal if its binary expansion is normal. We agree to ignore the integer part (since it has only finitely many digits, adding it as a prefix would not matter anyway).

[4] We go into all these details since the paper [32] is published (in Russian) in a quite obscure place: a volume in a series published by Moscow Pedagogical Institute. It seems that this volume is now (June 2019) missing even in the library of the very institute that published it (now it is called Moscow Pedagogical State University). Fortunately, this volume is available in the Russian State Library in Moscow (though it is included only in the paper cards version of the catalog, not in the electronic database).

Theorem 4 (Wall [46]). *If p and q are rational numbers and α is normal real number, then $\alpha p + q$ is normal.*

Proof. It is enough to show that a normal number remains normal when multiplied or divided by an integer (adding integers preserves normality for trivial reasons). Let N be some integer factor. Fix some positions $m, m+1, \ldots, m+k-1$ in the binary expansion. Look at the digits of reals α and $N\alpha$ that occupy these positions. They form two k-bit blocks, one for α and one for $N\alpha$. Knowing the first one, we have N possibilities for the second one (a school division algorithm keeps remainder modulo N), and vice versa (multiplication by N also has this property). So if α is normal and frequencies of blocks in α are correct, in $N\alpha$ (or α/N) each block appears at most N times more often. It remains to apply Piatetski-Shapiro's theorem. □

Remark. Wall's theorem also can be derived from the characterization of normality in terms of automatic complexity (Theorem 2), since division and multiplication are automatic transformations.

4.3 Sufficient condition for normality in terms of complexity

As we have mentioned, Champernowne [14] proved that the concatenation of the positional representations of all integers (in increasing order) is a normal sequence. (He considered decimal representations, not binary, but this does not make any difference.)

This result is a special case of the following simple observation, a sufficient condition for normality in terms of Kolmogorov complexity.

Theorem 5. *Let $x_1, x_2, x_3, \ldots$ be a sequence of non-empty binary strings. Let L_n be the average length of $x_1, \ldots, x_n$, i.e., $L_n = (|x_1| + \ldots + |x_n|)/n$. Let C_n be their average Kolmogorov complexity, i.e., $C_n = (\mathrm{C}(x_1) + \ldots + \mathrm{C}(x_n))/n$. Assume that $|x_n| = o(|x_1| + \ldots + |x_{n-1}|)$ and $L_n \to \infty$ as $n \to \infty$.*

If $C_n/L_n \to 1$ as $n \to \infty$, then the concatenated sequence $\varkappa = x_1x_2x_3\ldots$ is normal.

The first two assumptions are technical (and usually are easy to check); they guarantee that $|x_n|$ grow not too slow and not too fast. In this case the normality of concatenation is guaranteed if average *complexity* of strings x_i is close to their average *length*. Note that C_n is defined up to $O(1)$ additive term (the complexity function is defined with the same precision) and that $C_n \leqslant L_n + O(1)$.

Proof. Using Theorem 2, we need to prove, for an arbitrary fixed automatic description mode R, a lower bound $N - o(N)$ for the automatic complexity of the N-bit prefix of $x_1x_2x_3\ldots$. This prefix may end inside some x_i; we ignore the last incomplete block and consider maximal prefix of the form $x_1 \ldots x_M$ of length at most N. Due to the superadditivity property (Theorem 1, (k)) the automatic complexity of the N-bit prefix is at least $\mathrm{C}_R(x_1) + \ldots + \mathrm{C}_R(x_M)$ and is at least $\mathrm{C}(x_1) + \ldots + \mathrm{C}(x_M) - O(M)$, since Kolmogorov complexity is a lower bound for automatic complexity up to $O(1)$ additive term.

Due to the assumption $|x_n| = o(|x_1| + \ldots + |x_{n-1}|)$, the ignored incomplete part has length $o(N)$, so we may replace N in the desired lower bound by $|x_1| + \ldots + |x_M|$. It remains to note that the ratio

$$\frac{\mathrm{C}_R(x_1) + \ldots + \mathrm{C}_R(x_M)}{|x_1| + \ldots + |x_M|} \geqslant \frac{\mathrm{C}(x_1) + \ldots + \mathrm{C}(x_M) - O(M)}{|x_1| + \ldots + |x_M|} = \frac{C_M - O(1)}{L_M}$$

converges to 1 according to our assumptions ($L_M \to \infty$ and $C_M/L_M \to 1$). Here in the last step we divided the numerator and the denominator by M. ☐

We formulated Theorem 5 for the binary case, but both the statement and the proof can be easily adapted to arbitrary base.

For the Champernowne example x_i is the binary representation of i. The average length of $x_1, \ldots, x_n$, and even the maximal length, is obviously bounded by $\log n + O(1)$. As for the complexity, it is enough to note that all x_i are different, and the number of different strings of complexity at most u is $O(2^u)$. Therefore, the fraction of strings that have complexity at most $\log n - d$ among all strings $x_1, \ldots, x_n$ is $O(2^{-d})$. The series $\sum d2^{-d}$ converges, so the average complexity of $x_1, \ldots, x_n$ is at least $\log n - O(1)$, and $C_n/L_n \geqslant (\log n - O(1))/(\log n + O(1)) \to 1$. Other conditions of Theorem 5 are obviously true.

Champernowne's paper [14] contains some other results: Theorem I says that the concatenation of all *strings* in order of increasing lengths, i.e., the sequence

$$0\,1\,00\,01\,10\,11\,000\,001\,010\ \ldots$$

is normal. Theorem II says that it remains normal if every string is repeated μ times for some integer constant μ. Theorem III is the Champernowne's example we started with. Theorem IV considers a sequence where ith string is repeated i times. In all these examples the normality obviously follows from our Theorem 5. (For Theorem IV we need to note that a subset that has measure at most 2^{-d} according to the uniform distribution on $\{1, 2, \ldots, n\}$ has measure $O(2^{-d})$ if we change the distribution and let the probability of i be proportional to i.)

4.4 Copeland – Erdös' theorem

In addition to Theorems I–IV (see the previous section) Champernowne [14] gave some other examples of normal numbers (sequences), saying that they "need for their establishment tedious lemmas and an involved notation, [and] no attempt at a proof will be advanced". These examples are the sequences made of concatenated representations of (a) all composite numbers, (b) numbers $\lfloor \alpha n \rfloor$ for some positive real α, and (c) $\lfloor n \log n \rfloor$. In all these cases the normality easily follows from Theorem 5.

Champernowne also stated as a conjecture that the sequence made of decimal representation of *prime* numbers is normal. This conjecture was proven by Copeland and Erdös [15] who gave a sufficient condition for the sequence $x_1x_2x_3\ldots$ obtained by concatenating the positional representations of integers $x_1, x_2, \ldots$ to be normal. Let us state the Copeland – Erdös theorem and show that it is a direct consequence of Theorem 5.

Theorem 6 (Copeland – Erdös)**.** *Let $x_1, x_2, x_3, \ldots$ be a strictly increasing sequence of integers, and the number of terms x_i that are less than 2^m is at least $2^{m(1-o(1))}$. The the sequence $x_1x_2x_3\ldots$ (the concatenation of the positional representations) is normal.*

Proof. The assumption implies that the length of x_n is $(1+o(1))\log n$, and the conditions for the lengths are true for obvious reasons. The lower bound for complexity in the Champernowne example (Section 4.3) used only that all x_i are different. $\square$

4.5 Besicovitch's theorem

Besicovitch [6] has proven that the number obtained by the concatenation of all perfect squares in the increasing order is normal. This result is also a consequence of Theorem 5, but more detailed analysis is needed. We give a sketch of the corresponding argument. Let x_i be the binary representation of i^2 (we will say later what changes are needed for decimal representation). The length of x_i is about $2\log i$, while the (typical) complexity is the complexity of i, so C_n/L_n is close to $1/2$, not 1. To deal with this problem, let us divide the string x_i in two halves of equal length $x_i = y_iz_i$ and consider the most significant half and the least significant half separately. Of course, x_i may have odd length, then the lengths of y_i and z_i differ by 1, and the reasoning should be adapted. We do not go into these details.

Instead of using $\mathrm{C}(x_i)$ as a lower bound for $\mathrm{C}_R(x_i)$, we note that $\mathrm{C}_R(x_i) \geqslant \mathrm{C}_R(y_i) + \mathrm{C}_R(z_i)$ and then use $\mathrm{C}(y_i)$ and $\mathrm{C}(z_i)$ as the lower bounds for both summands. (In other words, we apply our criterion to a sequence of left and right halves of x_i.) For y_i we note that the most significant half of i^2 determines i almost uniquely (there are $O(1)$ possible values of i with the same most significant half). Indeed, if i is of order 2^k, then changing i by 1 changes i^2 by $2i+1$, and this is almost enough to change the most significant half of i^2 (i.e., the k most significant bits out of $2k$). One should also take into consideration the possibility that two halves have different lengths (by 1), but this gives only $O(1)$ new candidates.

For the least significant half z_i more complicated analysis is needed, since z_i does not determine i and sometimes many different values of i share the same z_i. For example, if i is a k-bit number whose $k/2$ least significant bits are zeros, then i^2 is a $2k$-bit number with k trailing zeros, so we have about $2^{k/2}$ different values of i that share the same z_i (all zeros). This happens rarely, as the following lemma shows:

Lemma 6.1. *The average Kolmogorov complexity of $x^2 \bmod 2^k$ taken over all x modulo 2^k is $k - O(1)$.*

Proof sketch. As we mentioned, complexity $\mathrm{C}(x^2)$ can be much less than $\mathrm{C}(x)$ (if x ends with $k/2$ zeros, the complexity of x^2 is zero while the complexity of x could be $k/2$). However, such a difference is possible only if x ends with many zeros. More precisely, we have $\mathrm{C}(x^2 \bmod 2^k) \geqslant \mathrm{C}(x) - O(\zeta(x))$ for k-bit string x, where $\zeta(x)$ is the number of trailing zeros in x (the maximal $u \leqslant k$ such that 2^u divides x). This is enough, since the expected value of $\zeta(x)$ for random x modulo 2^k is $O(1)$ (half of all numbers have at least one trailing zero, half of those have at least one additional trailing zero, etc., and the series converges).

To prove the bound let us rewrite it as $\mathrm{C}(x) \leqslant \mathrm{C}(x^2) + O(\zeta(x))$. To specify x, it is enough to specify x^2 and the ordinal number of x in the set of all residues with the same square. Therefore, it is enough to show that the number of residues y modulo 2^k such that $x^2 = y^2 \pmod{2^k}$ is bounded by $2^{O(z)}$ if x has z trailing zeros. Indeed, assume that x has z trailing zeros and $x^2 = y^2$ for some other y modulo 2^k. Then $x^2 - y^2 = (x-y)(x+y)$ is a multiple of 2^k, therefore $x - y$ is a multiple of 2^u and $x + y$ is a multiple of 2^v for some u, v such that $u + v = k$. Then $2x$ is a multiple of $2^{\min(u,v)}$, so $\min(u, v) \leqslant z + 1$. Then $\max(u, v) \geqslant k - z + O(1)$ (recall that $\min(u, v) + \max(u, v) = u + v$), so one of $x - y$ and $x + y$ is a multiple of $2^{k-z+O(1)}$, and each case contributes at most $2^{z+O(1)} = O(2^z)$ solutions for the equation $x^2 = y^2 \pmod{2^k}$. □

Remark. The statement of the lemma involves Kolmogorov complexity, but in fact we are interested in the Shannon entropy of x^2 where x is a random integer modulo 2^k. To make this statement formal, we need to switch to prefix complexity and note that a prefix-free code for x^2 gives a bound for the average prefix complexity and vice versa.

See also Section 5.5 for the dimension version of Theorem 5 (Theorem 14).

Question. Is it possible to generalize these arguments and prove the Davenport – Erdös result (replace the squaring in Besicovitch's theorem by a polynomial of higher degree)? One possible approach would be to estimate the entropies of k subsequent bits in $P(x)$ where x is a random integer in $0..2^k$.

4.6 Calude – Salomaa – Roblot question answered by Calude – Staiger – Stephan

In this section we use our tools to give a simple answer to a question posed by Calude, Salomaa and Roblot [11, Section 6] and answered in [12] by a more compicated arguement. In [11] the authors define a version of automatic complexity in the following way. A deterministic transducer (finite automaton that reads an input string and at each step produces some number of output bits) maps a description string to a string to be described, and the complexity of y is measured as the minimal sum of the sizes of the transducer and the input string needed to produce y; the minimum is taken over all pairs (transducer, input string) producing y. The size of the transducer is measured via some encoding, so the complexity function depends on the choice of this encoding. "It will be interesting to check whether finite-state random strings are Borel normal" [11, p. 5677]. Since normality is defined for infinite sequences, one probably should interpret this question in the following way: is it true that normal infinite sequences can be characterized as sequences whose prefixes has finite state complexity close to length?

This question got a negative answer in [12]. Here we show that the our tools can be used to provide a simple proof of this negative answer. More precisely, in one direction this approach works, but in the other direction it fails. To avoid confusion between different versions of automatic complexity, we denote the complexity defined in [11] by

CSR(x). It depends on the choice of the encoding of transducers, but the claim is true for every encoding, so we assume that some encoding is fixed and omit it in the notation.

Theorem 7 ([12]).

(a) *If a sequence $\alpha = a_0a_1\ldots$ is not normal, then there exist some $c < 1$ such that the* $\mathrm{CSR}(a_0\ldots a_{n-1}) < cn$ *for infinitely many n.*

(b) *There exists a normal sequence $\beta = b_0b_1\ldots$ such that*

$$\liminf \mathrm{CSR}(b_0\ldots b_{n-1})/n = 0.$$

Proof. To prove the first statement, we repeat the argument used to prove the first part of Theorem 2. Indeed, the block code constructed in that argument can be decoded by a transducer. This transducer had some description of fixed length, and then we add the length of the encoded string. For long prefixes the transducer part does not matter, since the transducer is fixed and the length of the prefix goes to infinity.

For the second part we construct an example of a normal sequence using the Champernowne's idea and Theorem 5. The sequence will have the form

$$\beta = (B_1)^{n_1}(B_2)^{n_2}\ldots$$

Here B_i is the concatenation of all strings of length i (say, in lexicographical ordering, but this does not matter), and n_i is a fast growing sequence of integers.

To choose n_i, let us note first that for a periodic sequence (of the form XY^∞) the CSR-complexity of its prefixes of the form XY^k is o(length). Indeed, we may consider a transducer that first outputs X, then outputs Y for each input bit 1. So $\mathrm{CSR}(XY^m) = m + O(1)$, and the compression ratio is about $1/|Y|$. To get o(length), we use Y^c for some constant c as a period to improve the compression.

Now consider the complexity/length ratio for the prefixes of β if the sequence n_i grows fast enough. Assume that $n_1, n_2, \ldots, n_k$ are already chosen and we now choose the value of n_{k+1}. We may use the bound explained in the previous paragraph and let $X = (B_1)^{n_1}\ldots(B_k)^{n_k}$ and $Y = B_{k+1}$. For large enough n_{k+1} we get arbitrarily small complexity/length ratio. (Note that good compression is guaranteed only for some prefixes; when increasing k, we need to switch to another transducer, and we know nothing about the length of its encoding. This corresponds to lim inf in our statement.)

It remains to apply Theorem 5 to show that for fast growing sequence $n_1, n_2, \ldots$ the sequence β is normal. We apply the criterion by splitting B_k into pieces of length k (so all strings of length k appear once in this decomposition of B_k). We already know that the average Kolmogorov complexity of the pieces in B_k is $k - O(1)$ (and the length of all pieces is k). This is enough to satisfy the conditions from Theorem 5 *if $x_1\ldots x_n$ ends on the boundary of the block B_k*. But in general we need also to consider the last incomplete group of blocks that form a prefix of some B_k. The total length of these blocks is bounded by $|B_k|$, i.e., by $k2^k$. We need this group to be short compared to the rest, and this will be guaranteed if n_{k-1} (the lower bound for the length of the previous part) is much bigger than $k2^k$. And we assume that n_k grow very fast, so this condition is easy to satisfy. Theorem 7 is proven. ∎

5 Finite state dimension and automatic complexity

If a sequence is not normal, we may ask how far it is from being normal. This is measured by the notion of finite state dimension introduced in [16]. This is a "finite state version" of the notion of effective dimension (as we have discussed in the Introduction). Finite state dimension of a binary sequence is a number between 0 and 1; it is an upper bound for effective Hausdorff dimension. This number equals 1 for normal sequences. In this section we extend some of the results proven for normal sequences to the case of arbitrary finite state dimension, and discuss the connections between finite state dimension and effective Hausdorff dimension.

We start (Section 5.1) by considering an equivalent definition of finite state dimension in terms of entropy rates from [9] and prove that one may as well use the non-aligned frequencies in this definition. In the next section (Section 5.2) we show that the equivalence between aligned and non-aligned blocks in the definition of finite state dimension requires change in the block size. Then (Section 5.3) we give a simplified proof of the result of Doty, Lutz and Nandakumar [17] saying that finite state dimension of a real number is not changed when the number is multiplied by a rational number (a dimension version of Wall's theorem), improving the bound for entropies of k-bit blocks, and give a simple example showing that this bound is tight. Then we prove the characterization of finite state dimension in terms of automatic complexity (Section 5.4, Theorem 11). Moreover, as a byproduct we get (Section 5.5) a "stateless" characterization of finite state dimension that does not mention at all finite state automata or Shannon entropy and uses superadditive upper bounds for Kolmogorov complexity (Theorem 12), and another stateless characterization that uses some calibration condition instead of Kolmogorov complexity (Theorem 13). Then we recall the original definition of finite state dimension in terms of finite state s-gales (Section 5.6) and use the tools from algorithmic information theory (a finite state version of a priori probability) to give a simple proof of equivalence between this definition and the others. In Section 5.7 we use martingales to provide simple proofs for the results of Agafonov (Theorem 17) and Schnorr – Stimm (Theorem 18). Finally, in Section 5.8 we note that some more general notion of finite state measure can also be used to characterize finite-state dimension and normality.

5.1 Entropy rates for aligned and non-aligned blocks

Consider a sequence $\alpha = a_0a_1a_2\ldots$, and some positive integer k. As in the definition of normality, we can cut the sequence α into k-bit blocks (aligned version), or consider all k-bit substrings of α (non-aligned version). Then we consider limit frequencies of these blocks. In this way we get some distribution on the set $\mathbb{B}^k$ of all k-bit blocks. We want to define the finite state dimension of α as the limit of the Shannon entropy of this distribution divided by k (when k goes to infinity).

The problem is that limit frequencies may not exist, so we should be more careful. For every N take the first N blocks of length k and choose one of them uniformly at random. In this way we obtain a random variable taking values in $\mathbb{B}^k$. Consider the Shannon entropy of this random variable. This can be done in aligned (a) and non-aligned (na) settings, so we get two quantities:

$$H^{\mathrm{a}}_{k,N}(\alpha)=H(\alpha_{kI}\ldots\alpha_{kI+k-1}),\qquad H^{\mathrm{na}}_{k,N}(\alpha)=H(\alpha_{I}\ldots\alpha_{I+k-1}),$$

where $I\in\{0,\ldots,N-1\}$ (the block number) is chosen uniformly at random, and H denotes the Shannon entropy of the corresponding random variable.

Then we apply $\liminf_N$ as $N\to\infty$ and define

$$H^{\mathrm{a}}_{k}(\alpha)=\liminf_{N\to\infty}H^{\mathrm{a}}_{k,N}(\alpha),\qquad H^{\mathrm{na}}_{k}(\alpha)=\liminf_{N\to\infty}H^{\mathrm{na}}_{k,N}(\alpha).$$

The following result says that both quantities $H^{\mathrm{a}}_k(\alpha)$ and $H^{\mathrm{na}}_k(\alpha)$, divided by the block length k, converge to the same value as $k\to\infty$, and this value can also be defined as $\inf_k H_k(\alpha)/k$ (both in aligned and non-aligned versions).

Theorem 8. *For every bit sequence α we have*

$$\lim_k\frac{H^{\mathrm{a}}_k(\alpha)}{k}=\inf_k\frac{H^{\mathrm{a}}_k(\alpha)}{k}=\lim_k\frac{H^{\mathrm{na}}_k(\alpha)}{k}=\inf_k\frac{H^{\mathrm{na}}_k(\alpha)}{k}.$$

The sequence α is normal if and only if this common value equals 1.

Definition 5. This common value of these four quantities is called the *finite state dimension of α.*

The original definition of finite state dimension [16] was different (see Section 5.6 below), and the equivalence between it and the aligned version of the definition given above was shown in [9]. See also Theorem 15 (part 2) below. (The equivalence between non-aligned and aligned versions seems to be new.)

Proof. There are two ways to prove Theorem 8. Here we give a proof that uses basic tools from information theory, such as Shearer-type inequalities. One can also prove this result using the characterization of finite state dimension in terms of automatic complexity. We sketch this proof later, see the remark at the end of Section 5.4 (p. 35).

Lemma 8.1. *For every α, every k, every $K\geqslant k$:*

$$\frac{H^{\mathrm{na}}_K(\alpha)}{K}\leqslant\frac{H^{\mathrm{a}}_k(\alpha)}{k}+O\left(\frac{k}{K}\right).$$

Lemma 8.2. *For every α, every k, every $K\geqslant k$:*

$$\frac{H^{\mathrm{a}}_K(\alpha)}{K}\leqslant\frac{H^{\mathrm{na}}_k(\alpha)}{k}+O\left(\frac{k}{K}\right).$$

Let us show how these two lemmas imply Theorem 8. Take the lim sup of the both sides of these inequalities as $K\to\infty$:

$$\limsup_{K\to\infty}\frac{H^{\mathrm{na}}_K(\alpha)}{K}\leqslant\frac{H^{\mathrm{a}}_k(\alpha)}{k},\qquad \limsup_{K\to\infty}\frac{H^{\mathrm{a}}_K(\alpha)}{K}\leqslant\frac{H^{\mathrm{na}}_k(\alpha)}{k}.$$

Since this holds for all k,

$$\limsup_{K\to\infty}\frac{H^{\mathrm{na}}_K(\alpha)}{K}\leqslant\inf_k\frac{H^{\mathrm{a}}_k(\alpha)}{k},\qquad \limsup_{K\to\infty}\frac{H^{\mathrm{a}}_K(\alpha)}{K}\leqslant\inf_k\frac{H^{\mathrm{na}}_k(\alpha)}{k}.$$

Obviously we also have:

$$\inf_k\frac{H^{\mathrm{a}}_k(\alpha)}{k}\leqslant\limsup_{K\to\infty}\frac{H^{\mathrm{a}}_K(\alpha)}{K},\qquad \inf_k\frac{H^{\mathrm{na}}_k(\alpha)}{k}\leqslant\limsup_{K\to\infty}\frac{H^{\mathrm{na}}_K(\alpha)}{K},$$

so all four quantities coincide and are equal to both $\lim_k\frac{H^{\mathrm{na}}_k(\alpha)}{k}$ and $\lim_k\frac{H^{\mathrm{a}}_k(\alpha)}{k}$.

It remains to prove Lemmas 8.1 and 8.2.

Proof of Lemma 8.1. Fix some sequence $\alpha=a_0a_1a_2\ldots$, and consider some integer N. Take $I\in\{0,1\ldots,N-1\}$ uniformly at random and consider a random variable

$$\xi=a_I\ldots a_{I+K-1}$$

whose values are K-bit strings. In other words, this random variable is a randomly selected non-aligned block among the first N ones. By definition, the entropy of ξ is $H^{\mathrm{na}}_{K,N}(\alpha)$. Let us look on aligned k-bit blocks covered by the block ξ (i.e., the aligned k-bit blocks inside $I\ldots I+K-1$). The exact number of these blocks may vary depending on I, but there are at least $m=\lfloor K/k\rfloor-1$ of them (if there were only $m-1$ complete blocks, plus maybe two incomplete blocks, then the total length would be at most $k(m-1)+2k-2=km+k-2$, but we have $K/k\geqslant m+1$, i.e., $K\geqslant km+k$). We number m first blocks from left to right and get m random variables $\xi_1,\ldots,\xi_m$ (defined at the same space $\{0,\ldots,N-1\}$). For example, ξ_1 is the leftmost aligned k-bit block of α in the interval $I\ldots I+K-1$. To reconstruct the value of ξ when all ξ_i are known, we need to specify the prefix and suffix of ξ that are not covered by ξ_i (including their lengths). This requires $O(k)$ bits of information, so

$$H^{\mathrm{na}}_{K,N}(\alpha)=H(\xi)\leqslant H(\xi_1)+\ldots+H(\xi_m)+O(k).$$

We will show that for each $s\in\{1,\ldots,m\}$ the distribution of the random variable ξ_s is close to the uniform distribution over the first $\lfloor N/k\rfloor$ aligned k-bit blocks of α. The standard way to measure how close are two distributions on the same set X is to measure the *statistical distance* between them, defined as

$$\delta(P,Q)=\frac{1}{2}\sum_{x\in X}\bigl|P(x)-Q(x)\bigr|.$$

We claim that (for each $s\in\{1,2,\ldots,m\}$) the statistical distance between the distribution of ξ_s and the uniform distribution on the first $\lfloor N/k\rfloor$ aligned blocks converges to 0 as $N\to\infty$. First, let us note that for a fixed aligned block its probability to become s-th aligned block inside a random nonaligned block is exactly k/N (there are k possible positions for a random non-aligned block when this happens). The only exception to this rule are aligned blocks that are near the endpoints, and we have at most $O(K/k)$ of them. When we choose a random aligned block, the probability to choose some position is exactly $1/\lfloor N/k \rfloor$, so we get some difference due to rounding. It is easy to see that the impact of both factors on the statistical distance converges to 0 as $N \to \infty$. Indeed, the number of the boundary blocks is $O(K/k)$, and the bound does not depend on N, while the probability of each block (in both distributions) converges to zero.[5] Also, since $m = N/k$ and $m' = \lfloor N/k \rfloor$ differ at most by 1, the difference between $1/m$ and $1/m'$ is of order $1/m^2$, and converges to 0 even if multiplied by m (the number of blocks is about m).

Now we use the continuity (more precisely, the uniform continuity) of the entropy function and note that all $m = \lfloor N/k \rfloor - 1$ random variables in the right hand side are close to the uniform distribution on first $\lfloor N/k \rfloor$ aligned blocks (the statistical distance converges to 0), so

$$\liminf_{N\to\infty} H^{\mathrm{na}}_{K,N}(\alpha) \leqslant (\lfloor K/k \rfloor - 1) \liminf_{N\to\infty} H^{\mathrm{a}}_{k,\lfloor N/k \rfloor}(\alpha) + O(k),$$

and dividing by K we get the statement of Lemma 8.1. $\square$

Proof of Lemma 8.2. Take $I \in \{0, 1 \ldots, N-1\}$ uniformly at random. We need an upper bound for $H^{\mathrm{a}}_{K,N}(\alpha)$, i.e., for $H(a_{KI} \ldots a_{KI+K-1})$. For that we use Shearer's inequality (see, e.g., [41, Section 7.2 and Chapter 10]). In general, this inequality can be formulated as follows. Consider a finite family of arbitrary random variables $\eta_0, \ldots, \eta_{m-1}$ indexed by integers in $\{0, \ldots, m-1\}$. For every $U \subset \{0, \ldots, m-1\}$ consider the tuple η_U of all η_u where $u \in U$. If a family of subsets $U_0, \ldots, U_{s-1} \subset \{0, \ldots, m-1\}$ covers each element of U at least r times, then

$$H(\eta_U) \leqslant \tfrac{1}{r} \left(H(\eta_{U_0}) + \ldots + H(\eta_{U_{s-1}}) \right).$$

In our case we have K variables $\eta_0, \ldots \eta_{K-1}$ that are individual bits in a random aligned K-bit block $a_{KI} \ldots a_{KI+K-1}$ (for random I), i.e. $\eta_0 = a_{KI}$, $\eta_1 = a_{KI+1}$, etc. The set U contains all indices $0, \ldots, K-1$, and the sets U_i contains k indices $i, i+1, \ldots, i+k-1$ (where operations are performed modulo K, so there are U_i that combine the prefix and suffix of a random K-bit block). Each η_i is covered k times due to this cyclic arrangement. In other words, the variable η_{U_i} is a substring of the random string $\eta_U = a_{KI} \ldots a_{KI+K-1}$ that starts from ith position and wraps around if there is not enough bits. There are $k-1$ tuples of this "wrap-around" type (block of length k may cross the boundary in $k-1$ ways). These tuples are not convenient for our analysis, so we just bound their entropy by k. In this way we obtain the following upper bound:

$$H^{\mathrm{a}}_{K,N}(\alpha) = H(a_{KI} \ldots a_{KI+K-1}) \leqslant \frac{1}{k} \left(\sum_{s=0}^{K-k} H(a_{KI+s} \ldots a_{KI+s+k-1}) + (k-1)k \right).$$

[5] More precisely, we should speak not about probability of a given block, since the same k-bit block may appear in several positions, but about the probability of its appearance in a given position. Formally speaking, we use the following obvious fact: if we apply some function to two random variables, the statistical difference between them may only decrease. Here the function forgets the position of a block.

Adding $k-1$ terms (adding some other entropies that replace the "wrap-around terms"), we increase the right hand side:

$$H^{\mathrm{a}}_{K,N}(\alpha) \leqslant \frac{1}{k}\left(\sum_{s=0}^{K-1} H(a_{KI+s}\dots a_{KI+s+k-1}) + (k-1)k\right).$$

Let us look at the variable $a_{KI+s}\dots a_{KI+s+k-1}$ in the right hand side for some fixed s. It has the same distribution as the random non-aligned k-bit block $a_J \dots a_{J+k-1}$ for uniformly chosen J in $\{0,\dots,NK-1\}$ conditional on the event "$J \bmod K = s$":

$$H(a_{KI+s}\dots a_{KI+s+k-1}) = H(a_J \dots a_{J+k-1} | J \bmod K = s).$$

The average of these K entropies (for $s = 0,\dots,K-1$) is the conditional entropy

$$H(a_J \dots a_{J+k-1} | J \bmod K)$$

that does not exceed the unconditional entropy. So we get

$$H^{\mathrm{a}}_{K,N}(\alpha) \leqslant \frac{1}{k}\left(K \cdot H^{\mathrm{na}}_{k,KN}(\alpha) + (k-1)k\right).$$

By taking $\liminf$ as $N \to \infty$ we obtain:

$$\frac{H^{\mathrm{a}}_K(\alpha)}{K} = \liminf_{N\to\infty} \frac{H^{\mathrm{a}}_{K,N}(\alpha)}{K} \leqslant \liminf_{N\to\infty} \frac{H^{\mathrm{na}}_{k,KN}(\alpha)}{k} + O\left(\frac{k}{K}\right).$$

However, $\liminf$ in the right hand side is taken over multiples of K and we want it to be over all indices. Formally, it remains to show that

$$\liminf_{N\to\infty} \frac{H^{\mathrm{na}}_{k,KN}(\alpha)}{k} = \liminf_{N\to\infty} \frac{H^{\mathrm{na}}_{k,N}(\alpha)}{k}$$

as the latter is by definition equal to $H^{\mathrm{na}}_k(\alpha)/k$. Indeed, the statistical distance between distributions on the first KN (non-aligned) blocks and the distribution on the first $KN+r$ blocks (where r the remainder modulo K) tends to zero since the first distribution is the second one conditioned on the event whose probability converges to 1 (i.e., the event "the randomly chosen block is not among the r last ones" whose probability is $KN/(KN+r)$). □

Theorem 8 is proven. □

5.2 Why do we need large blocks: a counterexample

In the previous section we have shown that aligned finite state dimension is equal to the non-aligned one. However, this argument uses different block sizes: we show that if $H^a_k(\alpha)/k$ is small, then $H^{na}_K(\alpha)/K$ is small for much larger K, and vice versa. This change in the block size is unavoidable, as the following example shows (when $k = 2$, no fixed K is enough):

Theorem 9.

(a) *For all k there exists an infinite sequence α such that $H_2^{\mathrm{na}}(\alpha) < 2$ and $H_m^{\mathrm{a}}(\alpha) = m$ for all $m \leqslant k$.*

(b) *For all k there exists an infinite sequence α such that $H_2^{\mathrm{a}}(\alpha) < 2$ and $H_m^{\mathrm{na}}(\alpha) = m$ for all $m \leqslant k$.*

Proof. (a) Consider all k-bit strings. It is easy to arrange them in some order $B_0, \ldots$ such that the last bit of B_i is the same as the first bit of B_{i+1}, for all i, and the last bit of the last block is the same as the first bit of the first block. For example, consider (for every $x \in \{0,1\}^{k-2}$) four k-bit strings $0x0, 0x1, 1x1, 1x0$ and concatenate these 2^{k-2} quadruples in arbitrary order.

Then consider a periodic sequence α with period $B_0 B_1 \ldots B_{2^k-1}$. Obviously all aligned k-bit blocks appear with the same frequency in α, so $H_k^{\mathrm{a}}(\alpha) = k$. However, for non-aligned bit blocks of length 2 we have two cases: this pair can be completely inside some B_i, or be on the boundary between blocks. The pairs of the first type are balanced (since we have all possible k-bit blocks), but the boundary pairs could be only 00 or 11 due to our construction. So the non-aligned frequency of these two blocks is $1/4 + \Omega(1/k)$, and for two other blocks we have $1/4 - \Omega(1/k)$, so $H_2^{\mathrm{na}}(\alpha) < 2$.

The only problem is that in this construction we do not necessarily have that $H_m^{\mathrm{a}}(\alpha) = m$ for all $m < k$, only for $m = k$. But this is easy to fix. Note that $H_k^{\mathrm{a}}(\alpha) = k$ implies $H_m^{\mathrm{a}}(\alpha) = m$ if m is a divisor of k. So we can just use the same construction with blocks of length $k!$ instead of k.

(b) Now let us consider a sequence constructed in the same way, but blocks $B_0, B_1, \ldots, B_{2^k-1}$ go in the lexicographical ordering. First let us note that all k-bit blocks have the same *non-aligned* frequencies in the periodic sequence with period $B_0 B_1 \ldots B_{2^k-1}$. (For aligned k-blocks it was obvious, but the non-aligned case needs some proof.) Indeed, consider some k-bit string U; we need to show that it appears exactly k times in the (looped) sequence $B_0 B_1 \ldots B_{2^k-1}$. In fact, it appears exactly once for each position modulo k. For example, it appears once among the blocks B_i. Why the same it true for some other position $s \bmod k$ where the $k - s$ first bits of U appear as a suffix of B_{i-1} and the last s bits of U appear as a prefix of B_i? Note that $(k-s)$-bit suffixes of $B_0, B_1, B_2, \ldots$ form a cycle modulo 2^{k-s}, so the first $k - s$ bits of U uniquely determine the *last* $k - s$ bits of B_i, whereas the first s bits of B_i are just written in the s-bit suffix of U.

This implies that non-aligned frequencies for all k-bit blocks are the same. Therefore, they are the same also for m-bit blockes for all $m \leqslant k$. This implies also that we may assume for the rest that k is odd.

Now let us consider *aligned* blocks of size 2. We will show that aligned frequency of the block 10 in the sequence $B_0 B_1 \ldots B_{2^k-1}$ is $1/4 - \Omega(1/k)$. Since k is odd (see above), when we cut our sequence into blocks of size 2, there are "border" blocks that cross the boundaries between B_i and B_{i+1}, and other non-border blocks. Each second boundary is crossed (between B_0 and B_1, then B_2 and B_3, and so on), so the border blocks *all have the first bit* 0. In particular, 10 never appears on such positions. This creates discrepancy of order $1/k$ for 10, and we should check that it is not compensated by non-boundary blocks. In the blocks B_i with even i we delete that last bit and cut the rest into bit pairs. After deleting the last bit we have all possible $(k-1)$-bit strings, so no discrepancy arises here. In the blocks B_i with odd i we delete the first bit, and then cut the rest into bit pairs. In the last pair the last bit is 1 (since i is odd), so once again we never have 10 here, as required (the other positions are balanced). □

5.3 Finite state dimension and Wall's theorem

Using the notion of finite state dimension, one can generalize Wall's theorem, as noted by Doty, Lutz and Nandakumar [17]]

Theorem 10 (Doty, Lutz, Nandakumar)**.** *The finite state dimension of a real number does not change when the number is multiplied by a rational number or when a rational number is added.*

Proof. To prove this result, Doty, Lutz and Nandakumar show that for every k the block entropy rates for k-bit blocks in a binary representation of a real number do not change significantly when a real number is multiplied by an integer. This implies the same for division, and adding integers is trivial, so the finite state dimension does not change when we multiply by rational numbers or add rational numbers. More precisely, they show that

$$|H_k^{\mathrm{a}}(\alpha) - H_k^{\mathrm{a}}(M\cdot\alpha)| \leqslant \log_2(M^2(s+1))$$

for every real α and every positive integer M, where s is the number of ones in the binary expansion of M. This inequality implies that finite state dimensions of α and $M\alpha$ are the same, since the bound does not depend on k and, being divided by k, converges to 0 as $k\to\infty$. In fact, a much more simple argument provides a better bound:

Lemma 10.1. *For any real α and any positive integer M:*

$$|H_k^{\mathrm{a}}(\alpha) - H_k^{\mathrm{a}}(M\cdot\alpha)| \leqslant \log_2 M \quad \textit{and} \quad |H_k^{\mathrm{na}}(\alpha) - H_k^{\mathrm{na}}(M\cdot\alpha)| \leqslant \log_2 M.$$

Proof. Both inequalities (aligned and non-aligned versions) are proven in a similar way. Consider, for instance, the aligned case. Choose $i\in\{0,\ldots,N-1\}$ uniformly at random and let X be the ith aligned k-bit block in α. Define a random variable Y for $M\cdot\alpha$ in a similar way.

As we have noted while proving Theorem 4, for each group of neighbor positions $(i, i+1, \ldots, i+k-1)$, the bits of α in these positions determine almost uniquely the bits of $M\alpha$ in the same positions, and vice versa. Here "almost uniquely" means that there are at most M possibilities. Therefore

$$H(X|Y)\leqslant \log_2 M \quad\text{and}\quad H(Y|X)\leqslant\log_2 M.$$

Since $H(X)\leqslant H(X,Y)=H(Y)+H(X|Y)$ and $H(Y)\leqslant H(X,Y)=H(X)+H(Y|X)$, we have

$$|H(X)-H(Y)|\leqslant\log_2 M.$$

□

As we have said, Lemma 10.1 immediately implies Theorem 10. $\square$

The bounds provided by Lemma 10.1 are sharp, as the following example shows. Note that

$$1/3 = 0.(01), \qquad 1/9 = 0.(000111),$$

(parentheses show the period of a periodic fraction), $H_6^{\mathrm{na}}(1/3) = \log_2 2, H_6^{\mathrm{na}}(1/9) = \log_2 6$, and $H_2^{\mathrm{a}}(1/3) = \log_2 1, H_2^{\mathrm{a}}(1/9) = \log_2 3$.

5.4 Finite state dimension and automatic complexity

The characterization of normal sequences in terms of automatic Kolmogorov complexity can be extended to the case of arbitrary finite state dimension.

Theorem 11. *Finite state dimension of an arbitrary bit sequence $\alpha = a_0a_1a_2\ldots$ is equal to*

$$\inf_R \liminf_{n\to\infty} \frac{\mathrm{C}_R(a_0a_1\ldots a_{n-1})}{n}$$

Note that replacing C_R by standard Kolmogorov complexity, we get the definition of the effective Hausdorff dimension (and $\inf_R$ is no more needed, since there exists an optimal description mode).

Proof. Let us give the proof sketch first. We need to prove two inequalities. In one direction we assume that finite state dimension of α is small (less than some τ) and need to construct an automatic description mode R such that $\liminf_{n\to\infty} \mathrm{C}_R(a_0a_1\ldots a_{n-1})/n$ is less than τ. The basic idea is simple: if for some k the distribution on aligned blocks has small entropy, then the corresponding Shannon – Fano code has small average coding length and therefore provides good compression ratio when used as a description mode. However, there are two problems with this plan. First, for different prefixes that have small entropy distributions, these distributions could be different, but we need to construct one description mode that provides good compression for infinitely many prefixes. Second, the Shannon–Fano code does not reach the exact value of entropy, the average length may exceed entropy by at most 1.

We have already seen in the proof of Theorem 2 how to deal with these problems. For the first one, we consider the sequence of distributions with small entropies, use compactness to choose a convergent subsequence, and construct the Shannon–Fano code using the limit distribution. For the second, we know that the difference between entropy and average length is at most 1, and this 1 is divided by the length of the block, so we may make the difference per bit arbitrarily small by considering blocks of large length. (In the proof of Theorem 2 we doubled the length of the blocks; now we may do the same or use a similar argument implicitly by using Theorem 8 that allows us to start with blocks of arbitrarily large length.)

For the second part we assume that finite state dimension of some sequence α is high. Then we fix some automatic description mode R and prove the lower bound for the automatic complexity C_R of the prefixes of α. For that we cut the prefix into aligned blocks of some size k and note that the complexity of the entire prefix is at least the sum of the block complexities. Then we note that one may use (with small distortion for large blocks) the prefix version of Kolmogorov complexity (see [41] for its definition and properties)[6]. The prefix version of complexity by definition determines some prefix-free code for the blocks, so the entropy of the blocks distribution provides a lower bound for the average prefix complexity of a block. It remains to note that the constant in the equality connecting C_R and C depends only on R, but not on the block size, and the difference between plain and prefix complexity is $O(\log k)$ for blocks of size k.

Now we give the details. For the first part we use the following statement (that was already used implicitly in the proof of Theorem 2):

Lemma 11.1. *Let k be some integer and let P be a distribution on a set $\mathbb{B}^k$ of k-bit blocks. Then there exists an automatic description mode R such that for every string x whose length is a multiple of k, we have*

$$\mathrm{C}_R(x) \leqslant \frac{|x|}{k}\left(\sum_B Q(B)\log\frac{1}{P(B)} + 1\right)$$

where Q is the distribution on k-bit blocks appearing when x is split into blocks of size k.

Proof of Lemma 11.1. Consider the Shannon–Fano code for k-bit blocks based on the distribution P. Then the length of the code for arbitrary block B is at most $\log(1/P(B))+1$. This code is prefix-free and can be uniquely decoded bit by bit. Therefore, it corresponds to some automatic description mode R. If a string x is a concatenation of k-bit blocks, then x has a description whose length is the sum of the lengths of the codes for these blocks, and we get the required bound. ☐

Note that in this lemma we allow some values $P(B)$ to be zeros; if such a block appears in x, the right hand side is infinite and the inequality is vacuous. Still we need to be careful (see the discussion below).

Now consider some sequence α whose finite state dimension (defined in terms of aligned block entropies) is smaller that some threshold τ. Choose some intermediate $\tau' < \tau$ such that finite state dimension of α is less than τ'. Recall that the finite state dimension can be defined as infimum over k, and choose some k such that $\liminf$ of entropies in the sequence of aligned k-blocks is less than τ'. By doubling the value of k and using the argument from Theorem 2 or using the statement of Theorem 8 we may assume without loss of generality that k is as large as we need (a side remark that will be used later). Since the $\liminf$ of the entropies per bit in the increasing prefixes is less than τ', there exists a sequence of prefixes of growing lengths such that all corresponding distributions of k-bit blocks all have entropy less than $k\tau'$. The set of all distributions on k-bit blocks is compact, therefore we may choose a subsequence of distributions Q_N

[6] If desired, the proof can be reformulated without references to Kolmogorov complexity theory; we will discuss this reformulation later.

that converges to some distribution P on $\mathbb{B}^k$, and $H(Q_N) < k\tau'$ for all N. The entropy is a continuous function on the set of all distributions, therefore $H(P) \leqslant k\tau'$.

Assume first that all values of P are positive (no zeros). Then we use Lemma 11.1 for the distribution P and get some automatic description mode R. The lemma provides an upper bound for the C_R-complexity of the prefixes that correspond to distributions Q_N. Since Q_N converge to P and $H(P) \leqslant \tau' k$, we conclude that for these prefixes we have

$$\liminf_{n\to\infty} \mathrm{C}_R(a_0 a_1 \ldots a_{n-1})/n \leqslant \tau' + 1/k.$$

For large enough k we have $\tau' + 1/k < \tau$, and the get the required statement.

If some values of P are zeros (some blocks have zero probability in the limit distribution), we have a problem: we cannot use the code provided by Lemma 11.1, since some blocks that have zero limit frequency, still appear in the prefixes of α, and they have no code. There are several ways to avoid this problem. For example, one may change the code provided by the Lemma, adding leading 0 to all codewords, and use codewords starting from 1 to encode "bad" blocks (having zero probability). Then all blocks, including bad ones, will have codes of finite length, the constant 1 in the Lemma is replaced by 2 (and this does not hurt), and we can proceed as before. (The exact lengths of codewords for bad blocks do not matter, since their limit frequency is zero.) The other possibility is to use some P' that is close to P and has all non-zero probabilities. Then the average length of code will be bigger, and the overhead is the Kullback – Leibler distance between P and P', so it can be made arbitrarily small, so it does not matter for $\liminf$.

The first part is proven.

For the second part, let us assume that for every k the $\liminf$ of entropies in the distributions of aligned k-blocks (per bit) is greater than τ. So, for a fixed k and for all sufficiently long prefixes the entropy of the corresponding distribution on aligned k-blocks is greater than τk. Fix some automatic description mode R. We will prove that for large enough prefixes $a_0 a_1 \ldots a_{n-1}$ the C_R-complexity per bit is large, namely, exceeds $\tau - O(k/n) - O(\log k/k)$. Taking $\liminf$ when $n \to \infty$, we get rid of $O(k/n)$; since k is arbitraryly large, we obtain the lower bound τ for $\liminf \mathrm{C}_R(a_0 \ldots a_{n-1})/n$, as required.

To get the lower bound for $\mathrm{C}_R(a_0 a_1 \ldots a_{n-1})$, we use that C_R is a superadditive upper bound for complexity. We state the corresponding lemma for arbitrary superadditive functions, since this generalization will be useful later.

Definition 6. A non-negative function F on strings is called *superadditive*, if $F(xy) \geqslant F(x) + F(y)$ for every two strings x and y.

Lemma 11.2. *Let $F(x)$ be a superadditive non-negative real-valued function on strings. Assume that F is an upper bound for Kolmogorov complexity with logarithmic precision, i.e.,*

$$\mathrm{C}(u) \leqslant F(u) + O(\log |u|)$$

for every string u. Then for every string x that is a concatenation of k-bit blocks for some k, if Q is the distribution on $\mathbb{B}^k$ that corresponds to the frequencies of these blocks in x, we have

$$F(x) \geqslant \frac{|x|}{k}\,(H(Q) - O(\log k))\,.$$

Proof of Lemma 11.2. Assume that x consists of m aligned k-bit blocks $B_0, \ldots, B_{m-1}$, i.e., $x = B_0 B_1 \ldots B_{m-1}$; here $m = |x|/k$. Then

$$\begin{aligned} F(x) = F(B_0 B_1 \ldots B_{m-1}) &\geqslant F(B_0) + \ldots + F(B_{m-1}) \geqslant \\ &\geqslant \mathrm{C}(B_0) + \ldots + \mathrm{C}(B_{m-1}) - O(m \log k) \geqslant \mathrm{K}(B_0) + \ldots + \mathrm{K}(B_{m-1}) - O(m \log k). \end{aligned}$$

First we use the superadditivity property, then the upper bound property, and then we switch from the plain Kolmogorov complexity to the prefix Kolmogorov complexity (see [44, 41]). The difference between plain and prefix complexity is $O(\log k)$ for k-bit strings, so it is absorbed by $O(m \log k)$ term.

Since prefix complexity determines some prefix-free code for all strings (and, in particular, for b-bit blocks), we may continue the sequence of inequalities:

$$F(x) \geqslant \mathrm{K}(B_0) + \ldots + \mathrm{K}(B_{m-1}) - mO(\log k) \geqslant mH(Q) - mO(\log k),$$

where $H(Q)$ is the Shannon entropy of the distribution Q. This is exactly the statement of the lemma. □

Now we apply the lemma to get a lower bound for $\mathrm{C}_R(a_0 a_1 \ldots a_{n-1})$ for an arbitrary prefix $a_0 \ldots a_{n-1}$ of α. We let $F = \mathrm{C}_R$, use arbitrary k and let $m = \lfloor |x|/k \rfloor$. We split the prefix $a_0 a_1 \ldots a_{n-1}$ into m blocks of length k (deleting the last incomplete block) and use the bound provided by the lemma. Dividing by n, we get

$$\frac{\mathrm{C}_R(a_0 a_1 \ldots a_{n-1})}{n} \geqslant \tfrac{m}{n}(H(Q) - O(\log k)) \geqslant \tfrac{m}{n}(\tau k - O(\log k)) \geqslant \tau - \frac{O(k)}{n} - \frac{O(\log k)}{k}.$$

Here Q is the distribution on k-blocks whose entropy by our assumption is at least τk; the last step is valid since $n = km + O(k)$. So we get the desired inequality, and this finishes the proof of Theorem 11. □

Remark. In fact C_R is an upper bound for C with $O(1)$-precision; we use a weaker bound $O(\log k)$ in the assumption of Lemma 11.2 since it is enough and will be used later when Lemma 11.2 is applied to some other upper bound for complexity.

One may wish to get rid of Kolmogorov complexity notions in this proof (used in the second part), and use only information-theoretic notions like entropy (this is unavoidable, since the statement of Theorem 11 involves entropy). Let us sketch this modified proof. Lemma 11.2 should be replaced by the following version:

Lemma 11.3. *Let $F(x)$ be a superadditive non-negative real-valued function on strings, i.e., $F(xy) \geqslant F(x) + F(y)$ for every two strings x and y. Assume that F satisfies the following* calibration condition:

$$\sum_{|x|=k} 2^{-F(x)} = O(\mathrm{poly}(k))$$

for every length k. Then for every string x that is a concatenation of k-bit blocks for some k, if Q is the distribution on $\mathbb{B}^k$ that corresponds to the frequencies of these blocks in x, we have

$$F(x) \geqslant \frac{|x|}{k}\left(H(Q) - O(\log k)\right).$$

Here $\mathrm{poly}(k)$ denotes some polynomial in k, i.e., we assume the polynomial growth of this sum (taken over all k-bit strings x).

Proof. Again, we split x into $m = |x|/k$ blocks of length m, so $x = B_0B_1 \ldots B_{m-1}$. Then

$$F(x) = F(B_0B_1 \ldots B_{m-1}) \geqslant F(B_0) + \ldots + F(B_{m-1}),$$

and we need to get a lower bound for the sum in the right hand side. For that, note that for a integer-valued function $F'(x) = \lfloor F(x) + c\log|x| \rfloor$ we have

$$\sum_{|x|=k} 2^{-F'(x)} \leqslant 1,$$

since the $c\log|x|$ additive term in the exponent compensates $O(\mathrm{poly}(k))$, so there exists a prefix code for k-bit strings where the codeword for a string x has length $F'(x)$. The average length of this code for k-blocks distributed according to Q is at least $H(Q)$, so we have

$$\frac{F'(B_0) + \ldots + F'(B_{m-1})}{m} \geqslant H(Q).$$

Therefore,

$$F(x) \geqslant F(B_0)+\ldots+F(B_{m-1}) \geqslant F'(B_0)+\ldots+F'(B_{m-1}) - mO(\log k) \geqslant m(H(Q) - O(\log k)),$$

as claimed. ☐

To finish the proof, we need to show that this lemma can be applied to $F = \mathrm{C}_R$ for arbitrary R. We may assume without loss of generality that $\mathrm{C}_R(x) \leqslant |x| + O(1)$, due to Theorem 1 (ah). Then every string x of length k has a description of length at most $k + O(1)$ (according to R). Then

$$\sum_{|x|=k} 2^{-\mathrm{C}_R(x)}$$

does not exceed the sum of $2^{-|u|}$ for all description of size at most $k_O(1)$. Indeed, take for every x its shortest description u; the restriction $|x| = k$ is not important, we only need to know that the shortest description has length at most $k + O(1)$. And this sum is $k + O(1)$, since each length gives 1.

Remark. In fact, our proof of Theorem 11 gives a bit more that we claimed. Namely, in this proof we prove the inequalities between the quantities used in the two definitions in the strongest possible form: we show that

- If the block entropy is small for some k, then infinitely many prefixes have small automatic complexity. To prove this, we may use the limit distribution for k-bit blocks (or $2k$-bit blocks, or $4k$-bit blocks, etc.) to construct a code that provides good compression ratio.

- If the block entropy is large for some large k, then all prefixes have large automatic complexity. For that we split the sequence into k-bit blocks, use superadditivity and provide a bound for compression ratio (with error $O(\log k/k)$)

These two arguments show that

$$\limsup_k H^{\mathrm{a}}_k(\alpha) \leqslant \inf_R \liminf_n (\mathrm{C}_R(a_0 \ldots a_{n-1})/n) \leqslant \inf_k H^{\mathrm{a}}_k(\alpha),$$

so we conclude that

$$\lim_k H^{\mathrm{a}}_k(\alpha) = \inf_k H^{\mathrm{a}}_k(\alpha)$$

(and the limit exists) without using Theorem 8. Moreover, we can adapt the proof of Theorem 11 for non-aligned blocks to prove a similar equality for the non-aligned case, thus deriving the full statement of Theorem 8 without using information-theoretic arguments like Shearer-type inequality. Let us sketch this argument.

In one direction we assume that the non-aligned block entropy for some block size k is small, and want to show that infinitely many prefixes are compressible enough. Note that the distribution of non-aligned k-blocks is the average of k distributions that correspond to aligned blocks in the original sequence α, then α without the first bit, then α without two first two bits, etc. The average of the entropies of these distributions is smaller than the entropy of the average distribution (convexity of entropy; we discussed it for the case of two distributions), so one of these k sequences has compressible prefixes. The deleted bits then can be added back without changing much the automatic complexity, so the original sequence α is also compressible.

In the other direction things are a bit more complicated. We can get a lower bound for the automatic complexity by splitting the sequence into k-bit blocks, but this lower bound involves the entropy of the aligned distribution. Of course, we can shift the boundaries modulo k, and get another lower bound for the same automatic complexity that involves another aligned distribution (for the sequence without first 0, 1,..., or $k-1$ bits). Averaging this bounds, we get a boundary that involves the average of the entropies of these k distributions. (We can take maximum instead of the average, but we will not need this.) The problem, however, is that this average may be smaller than the entropy of the non-aligned distribution (that is the average of k aligned distributions). More precisely, this average entropy is the *conditional* entropy of the nonaligned distribution when the condition is the position of the block modulo k. It remains to note that the difference between unconditional and conditional entropy is bounded by the entropy of the condition, i.e., $\log k$. Since we study the entropy per bit and divide the entropy by k, this difference does not matter ($\log k/k \to 0$).

5.5 Machine-independent characterization of normal sequences and finite state dimension

In fact we have proven the following characterization of finite state dimension that does not mention explicitly finite automata or entropies.

Theorem 12. *Let $\alpha = a_0a_1\ldots$ be an infinite bit sequence. Then the finite state dimension of α is the infimum*

$$\inf_F \liminf_n \frac{F(a_0a_1\ldots a_{n-1})}{n}$$

taken over all superadditive functions F that are upper bounds of Kolmorogov complexity with logarithmic precision, i.e., $\mathrm{C}(x) \leqslant F(x) + O(\log|x|)$ for all x.

Proof. First, we may use C_R (or KA_R defined in the next section) as F, and this shows that the $\inf_F$ in question does not exceed the finite state dimension. The inequality in the other direction is already proven since we used only these properties (the superadditivity and the upper bound property) in the proof of Theorem 11). □

One can say that this result explains the intuitive meaning of finite state dimension: it measures the compressibility of prefixes of α if only "local" compression/decompression methods are allowed where splitting the uncompressed sequence induces splitting of its compressed version.

Remark. This theorem is quite robust:

- We can replace the term $O(\log|x|)$ by $O(1)$, since the function C_R used in the proof does not exceed $\mathrm{C}(x) + O(1)$.
- We can also replace the term $O(\log|x|)$ by $o(|x|)$, since it is enough for the proof of the lower bound for $\liminf F(a_0\ldots a_{n-1})/n$.

One can delete all references to Kolmogorov complexity replacing them by the calibration condition saying that there are not too many strings with small F-values.

Theorem 13. *Let $\alpha = a_0a_1\ldots$ be an infinite bit sequence. Then the finite state dimension of α is the infimum*

$$\inf_F \liminf_n \frac{F(a_0a_1\ldots a_{n-1})}{n}$$

taken over all superadditive functions F such that

$$\sum_{|x|=k} 2^{-F(x)} = O(\mathrm{poly}(k))$$

Proof. No new argument is needed, since only this calibration condition was used in the proof, and the function C_R satisfies this calibration condition (as we have shown). □

Remark. We can replace the calibration condition by the other one (that is satisfied by the plain Kolmogorov complexity function): the number of strings x such that $F(x) < m$, is $O(2^m)$. Indeed, the function C_R satisfies this condition. On the other hand, it implies the condition used in Theorem 12. Indeed, assume that there is at most $O(2^m)$ strings x with $F(x) < m$. We want to prove that

$$\sum_{|x|=k} 2^{-F(x)} = O(k)$$

Indeed, there are at most 2^m strings with $F(x) = m$, so their sum is at most $O(1)$ for every m. We sum up all this bounds for $m < k$, and separately note that all strings x with $F(x) \geqslant k$ have $2^{-F(x)} \leqslant 2^{-k}$, and there are at most 2^k terms in the sum, so their sum is at most 1.

Later (Section 5.6, remark after Theorem 16) we will see another calibration condition that can be used in the theorem: $\sum_{x\in P} 2^{-F(x)} \leqslant c$ for some c and for every prefix-free set P. It obviously implies the condition used in Theorem 12, so we need only to provide a proof in the other direction, i.e., show a superadditive function that satisfies this condition and can be used in the proof instead of C_R. This function will be a finite-state version of a priori probability (maximal continuous semimeasure in the Kolmogorov complexity theory, see [41]).

The characterization of finite state dimension in terms of automatic complexity allows us to extend the sufficient condition for normality (Section 4.3, Theorem 5) and get the following lower bound for finite state dimension.

Theorem 14. *Let $x_1, x_2, x_3, \ldots$ be a sequence of non-empty binary strings. Let L_n be the average length of $x_1, \ldots, x_n$, i.e., $L_n = (|x_1| + \ldots + |x_n|)/n$. Let C_n be their average Kolmogorov complexity, i.e., $C_n = (\mathrm{C}(x_1) + \ldots + \mathrm{C}(x_n))/n$. Assume that $|x_n| = o(|x_1| + \ldots + |x_{n-1}|)$ and $L_n \to \infty$ as $n \to \infty$. Then the finite state dimension of the bit sequence $\varkappa = x_1x_2x_3\ldots$ is at least* $\liminf C_n/L_n$.

Proof. Using the characterization in terms of automatic complexity, we need to show that for every automatic description mode R the liminf of $\mathrm{C}_R(u)/|u|$, where u is a prefix of $\varkappa$, is at least $\liminf_n C_n/L_n$. If u ends on the block boundary, i.e., if $u = x_1 \ldots x_n$ for some n, then

$$F(u) = F(x_1 \ldots x_n) \geqslant F(x_1) + \ldots + F(x_n) \geqslant \mathrm{C}(x_1) + \ldots + \mathrm{C}(x_n) - O(n) = nC_n - O(n),$$

since $\mathrm{C}(x) \leqslant \mathrm{C}_R(x) + O(1)$. At the same time $|u| = nL_n$, so

$$\mathrm{C}_R(u)/|u| \geqslant C_n/L_n - O(1)/L_n.$$

That gives the desired bound for prefixes that end on the block boundaries.

Now we should consider u that do not end on the block boundary. We can delete the last incomplete block and get slightly shorter u'. For this u' we use the same bound as before, and due to the superadditivity it works as a bound for u. However, we have

$|u|$ in the denominator, not $|u'|$. This does not change the lim inf, since we assume that $|x_n| = o(|x_1| + \ldots + |x_{n-1}|)$, so the length of the incomplete block is negligible compared to the total length of previous complete blocks, and the correction factor converges to 1. Theorem 14 is proven. □

5.6 Finite state martingales and automatic a priori complexity

The original definition of finite-state dimension [16] was given in terms of games (or martingales corresponding to games). In this section we review this definition and show that it is equivalent to the definitions given above. This equivalence was proven in [9]; we provide a simple proof based on a finite-state version of the notion of a priori probability. But let us first say a few general words about the game approach to randomness that goes back to Ville (see his book [45]; see [7] for more historic details).

The game approach to randomness is based on the following idea: a bit sequence is not random if we can become infinitely rich playing against this sequence. The game is as follows: before seeing the next bit of the sequence, we have some amount of money m and split it into two parts $m = m_0 + m_1$, making two bets (on zero and one). Then the next bit is shown, the wrong bet is lost and the correct bet is doubled.[7] So our capital after seeing the next bit b is $2m_b$. The strategy in such a game is a function saying how we should split our capital after seeing a prefix of the sequence we are playing with. The usual way to describe the strategy is to provide the corresponding martingale, a non-negative real-valued function $m(x)$ that says what is our capital after playing with prefix x. It is easy to see that the rules of the game described above mean that

$$m(x) = \frac{m(x0) + m(x1)}{2} \tag{$*$}$$

for every string x.

Definition 7. A *martingale* is a function m with non-negative values defined on all binary strings that satisfies the equality $(*)$ for all x.

A more general notion of martingale is used in the probability theory, but for our purposes this special case is enough. After playing with prefix x, we split the capital $m(x)$ and make bets $m(x0)/2$ and $m(x1)/2$; the correct bet is doubled and our capital becomes $m(x0)$ or $m(x1)$.

Definition 8. A martingale *wins* on a binary sequence α if it is not bounded on the prefixes of α.

Martingales are in one-to-one correspondence with measures on the Cantor space. A measure on the Cantor space is determined by its values on *intervals* $[x]$; here $[x]$ is an interval that contains all sequences that have prefix x. For a measure μ we have $\mu([x]) = \mu([x0]) + \mu([x1])$ (the measure of the entire space is 1). The uniform Lebesgue measure

[7] One may wish to keep some part of the capital not using it for bets, but the same result can be achieved by betting half of it on zero and half of it on one.

λ on the Cantor space is defined as $\lambda([x]) = 2^{-|x|}$ and corresponds to independent fair coin tosses. The following statement follows directly from the definitions:

Proposition 2. *If μ is some measure on the Cantor space of bit sequences, then*

$$m(x) = \frac{\mu([x])}{\lambda([x])}$$

is a martingale. Every martingale corresponds to some measure in this way.

The conditional probabilities $\mu([x0])/\mu([x])$ and $\mu([x1])/\mu([x])$ are the fractions of capital that are used to bet on 0 and 1 respectively, after seeing the prefix x.

There is a characterization of Martin-Löf random sequences in terms of martingales [37]: a sequence is Martin-Löf random if and only if no lower semicomputable martingale wins against this sequence. (Lower semicomputability means that there is an algorithm that, given a string x, produces an increasing computable sequence of rational numbers that converges to the martingale value $m(x)$.) A "scaled-down" version of this result [38] says that a sequence is normal if and only if no finite state martingale wins against it. Informally speaking, finite state martingales correspond to strategies with finite memory, i.e., the strategies that have finite number of states, each state determines the proportion of bets (a pair of rational numbers whose sum is 1), and the next state is determined by a previous state and the bit seen). We will prove this result in Section 5.7 (Theorem 18).

Martingales can also be used to define effective Hausdorff dimension and (in the case of finite state martingales) finite state dimension [24, 25]. The dimension of a bit sequence determines how fast a martingale can grow on the prefixes of this sequence: dimension is the $\liminf$ of s such that $m(x)/2^{(1-s)|x|}$ is unbounded for some martingale m. Technically it is convenient to introduce the notion of *s-gale* for $s \in [0, 1]$.

Definition 9. Let $s \in [0, 1]$. An s-gale is a function $m(x)$ on binary strings with non-negative real values such that

$$m(x) = \frac{m(x0) + m(x1)}{2^s}$$

This definition introduces a "tax": after each game the capital is multiplied by factor 2^{s-1}. For $s = 1$ we have no tax: 1-gales are just martingales. For $s = 0$ we have tax rate 50% (half of the capital is taken away after each game). In the latter case ($s = 0$) we cannot win: even if we guess all the bits correctly and make the corresponding bets, our capital will only remain unchanged.

It is easy to see that we may equivalently define s-gales as functions of type

$$m(x) = \frac{\mu([x])}{2^{-s|x|}}$$

where μ is some measure.

The effective Hausdorff dimension of a bit sequence $\alpha = a_0a_1\ldots$ can be defined as infimum of the values of s such that some lower semicomputable s-gale wins on α.

One can also equivalently define it as $\liminf \mathrm{C}(a_0 a_1 \ldots a_{n-1})/n$ (see [24, 25, 26] or [41, Sections 5.8 and 9.10] for a survey). As we have said, in [16] a parallel theory was developed for the finite state case. It used finite state gales to define the finite state dimension; then in [9] the equivalence between the definitions of finite state dimension in terms of gales and entropy rates was proven. We used the reverse order: the definition of finite state dimension was given in terms of the entropy rates, and now we are going to give a simple proof of this equivalence.

First let us give the exact definitions. Assume that a finite set of nodes (states) is given; one of them is called an initial state. For each node (state) there are two outgoing edges labeled $(0, p_0)$ and $(1, p_1)$, where p_0 and p_1 are non-negative rational numbers and $p_0+p_1 = 1$. This labeled graph (together with the initial state) determines a probabilistic process: it starts in the initial state and then changes the state in a natural way: an outgoing edge is selected (with probability written on it, i.e., the probability to choose an edge is the second component of a pair on that edge), and the corresponding bit (the first component of the pair) is sent to the output. We get a measure on the Cantor space.

Definition 10. Measures of this type are called *finite state measures*. If μ is a finite state measure on the Cantor space, the ratio $\mu([x])/2^{-s|x|}$ is called a *finite state s-gale*. A finite state 1-gale is called *finite state martingale*.

Schnorr and Stimm [38] introduced finite state martingales under the name (in German) "Vermögenfunktionen erzeugt von endlichen Automaten". Dai, Lathrop, Lutz and Mayordomo in their paper [16] use the name "1-account finite state s-gale" for the notion we consider. It is easy to see that finite state martingales correspond to the gambling strategies described above (the gambler's decision how to split the capital between two bets is computed by a finite automaton: nodes are state of this automaton, the rational numbers on the outgoing edges determine the bets, and the endpoints of the edges are next states for two possible values of the next bit).

Theorem 15.

1. (*Schnorr and Stimm,* [38, Satz 4.1]) *A sequence is normal if and only if no finite state martingale wins against it.*

2. (*Dai, Lathrop, Lutz, Mayordomo,* [16][8]) *The finite state dimension of α is the infimum of $s \in [0,1]$ such that there exists a finite state s-gale that wins against α; if there is no such s, the finite state dimension is* 1.

We start by proving the second statement. It gives immediately one implication in the first statement: if no martingale wins on it, no s-gale (for $s \in [0,1]$) can win on it, the finite state dimension is 1 and the sequence is normal. We postpone the proof of the reverse implication in the first part, since it uses some additional technique that goes back to Agafonov [1]. See below Theorem 18 for this implication.

[8]As we have said, in [16] this property was used as a definition of finite state dimension, so they formulated the result as the characterization of finite state dimension in terms of aligned entropy rate.

To prove Theorem 15, we follow the proof of Theorem 11 with some changes. Namely, we replace the notion of automatic complexity (as defined in Section 2) by a similar notion that resembles a priori complexity (logarithm of the maximal continuous semimeasure as defined in the algorithmic information theory, see [41] for details).

Definition 11. Let R be a finite-state probabilistic process (i.e., a labeled graph of described type), with initial state i, and let $\rho_{R,i}$ be the corresponding measure on the Cantor space. Its logarithm can be considered as a complexity measure, and we define

$$\mathrm{KA}_{R,i}(x) = -\log \rho_{R,i}([x])$$

for a binary string x. We can change the initial state (node) i, keeping the graph of R unchanged otherwise. Then we take minimum over all nodes i and define

$$\mathrm{KA}_R(x) = \min_i \mathrm{KA}_{R,i}(x);$$

In other words, we consider the maximal probability over all initial states, and its negative logarithm. This is technically important to get a superadditive function. (For similar reasons we did not fix an initial state when defining the automatic complexity.)

The following result, analogous to Theorem 11, is essentially a reformulation of the second part of Theorem 15.

Theorem 16. *Finite state dimension of an arbitrary bit sequence $\alpha = a_0a_1a_2\dots$ is equal to*

$$\inf_R \liminf_{n\to\infty} \frac{\mathrm{KA}_R(a_0a_1\dots a_{n-1})}{n}$$

Before proving this result, let us show that the second part of Theorem 15 follows from it. For that we need to show two things:

- if $s > \liminf_{n\to\infty}(\mathrm{KA}_R(a_0a_1\dots a_{n-1})/n)$ for some R, then there is an s-gale that wins against α;
- if there is an s-gale that wins against α, then $s \geqslant \liminf_{n\to\infty}(\mathrm{KA}_R(a_0a_1\dots a_{n-1})/n)$ for some R.

In both case we consider a probabilistic process R that corresponds to the s-gale. Note that $s > \mathrm{KA}_R(a_0a_1\dots a_{n-1})/n$ by definition means that $\rho_i([a_0a_1\dots a_{n-1}]) > 2^{-sn}$ for some initial state i. So if this happens for some R and infinitely many n (as it should if s exceeds $\liminf$), then we can select i that appears infinitely often, and for that i the corresponding s-gale exceeds 1 infinitely often. It does not mean winning according to our definition, but we can slightly decrease s first (in such a way that it still exceeds the $\liminf$) and apply the argument to the smaller gale (that we change s back and convert a sequence that exceeds 1 infinitely often into an unbounded sequence).

On the other hand, if some s-gale wins against α, then the corresponding distribution infinitely often exceeds 2^{-sn} for n-bit prefixes of α, and $\mathrm{KA}_R(a_0a_1\dots a_{n-1})$ is smaller than sn for infinitely many prefixes. This finishes the derivation of the second part of Theorem 15 from Theorem 16. Now let us prove Theorem 16.

Proof of Theorem 16. We follow the scheme used for the proof of Theorem 11 with minimal changes. In the first part, we need to prove the version of Lemma 11.1 where C_R is replaced by KA_R. For that we may consider the same prefix code from Shannon's theorem, and consider a probabilistic process that tosses a fair coin to choose the next bit of a growing string, and decodes this string with respect to the prefix code chosen (when the codeword's end is reached, we output the encoded string; this is done using several states where the next move is deterministic, i.e., has probability 1). The probability to get some string x as the output is at least 2^{-m} if m is the length of its description, so we get the same bound as in Lemma 11.1.

In fact, here a simpler argument is possible, we do not really need the coding argument (and corresponding +1 overhead). Instead, we may consider a finite state probabilistic process that generates probability distribution P (from Lemma 11.1) on the consecutive blocks (different blocks are independent) and get directly the inequality

$$\mathrm{KA}_R(x) \leqslant \frac{|x|}{k}\left(\sum_B Q(B)\log\frac{1}{P(B)}\right)$$

without the term "+1". Still one small problem remains: by definition, we want the transitional probabilities in the finite state process be rational numbers, so we need to replace P by some rational approximation, and again an (arbitrariraly small) additive term appears instead of "+1". In this way we also make all probabilities in the approximate distribution strictly positive (this is needed to have KA_R finite).

For the other direction, we apply Lemma 11.2 to the function $F(x) = \mathrm{KA}_R(x)$. This gives the desired result immediately, and it remains to show that function $\mathrm{KA}_R(x)$ has the required properties:

Lemma 16.1. *For every R the function $\mathrm{KA}_R(x)$ is a superadditive function that is an upper bound for Kolmogorov complexity up to $O(\log|x|)$ terms.*

Proof. To get the lower bound for $\mathrm{KA}_R(uv)$ we need to prove an upper bound for $\max_i \rho_i(uv)$ (where maximum is taken over all nodes of R). But $\rho_i(uv) = \rho_i(u)\rho_j(v)$ where j is the state where the process is after generating u, and this gives the required bound. Here it is essential that we take maximum of $\rho_i(x)$ over all i (minimum of $\mathrm{KA}_{R,i}(x)$ over all i), this is a technical trick that makes KA_R superadditive.

To show that $\mathrm{KA}_R(x)$ is an upper bound for plain complexity with $O(\log|x|)$ precision, we note that it is enough to show this for $\mathrm{KA}_{R,i}$ for each i, since we have finite number of possible i (that does not depend on x, only on R). And $\mathrm{KA}_{R,i}$ is an upper bound for a priori complexity KA (see [41, Section 5.3] for definition and properties), since R defines a computable measure on the Cantor space. It remains to note that a priori complexity and plain complexity of an arbitrary string x differ at most by $O(\log|x|)$ [41, Section 6.2]. □

This lemma finishes the proof of Theorem 16, and therefore the second part of Theorem 15 is also proven. We will return to the first part after discussing Agafonov's result (see Theorem 18 below). □

Remark. The function $F = \mathrm{KA}_R$ is a superadditive function that satisfies the following calibration property: there exists a constant c such that $\sum_{x\in P} 2^{-F(x)} \leqslant c$ for all prefix-free sets P. Indeed, $2^{-\mathrm{KA}_R(x)} = \max_i \rho_{R,i}(x)$, where i is one of the states of the labeled graph R. For a fixed i the function $\rho_{R,i}(x)$ is a measure, and the sum over all x from a prefix-free set is at most 1. So we may let c be the number of states in R (indeed, $\max_i$ does not exceed $\sum_i$).

This property implies the calibration condition in Theorem 13, so we may apply this theorem and avoid all references to different versions of Kolmogorov complexity.

5.7 Agafonov's and Schnorr–Stimm's theorems

In this section we derive another classical result about normal numbers, Agafonov's theorem [1], from the martingale characterization.

Agafonov's result is motivated by the von Mises' approach to randomness (see, e.g., [41, Chapter 9] for a historic account). As von Mises have mentioned, a random sequence (he used German word *Kollektiv*) should remain random after using a reasonable selection rule. More precisely, assume that there is some set S of binary strings. This set determines a "selection rule" that selects a subsequence from every binary sequence α. The selection works as follows: we observe a binary sequence $\alpha = a_0a_1a_2\ldots$ and select terms a_n such that $a_0a_1\ldots a_{n-1} \in S$ (without reordering the selected terms). We get a subsequence; if an initial sequence is "random" (is plausible as an outcome of a fair coin tossing), said von Mises, this subsequence should also be random in the same sense. The Agafonov's theorem says that for regular (automatic) selection rules and normality as randomness this property is indeed true.

Theorem 17 (Agafonov)**.** *Let $\alpha = a_0a_1a_2\ldots$ be a normal sequence. Let S be a regular (=recognizable by a finite automaton) set of binary strings. Consider a subsequence σ made of terms a_n such that $a_0a_1\ldots a_{n-1} \in S$ (in the same order as in the original sequence). Then σ is normal or finite.*

Proof. We already know that a sequence is *not* normal if and only if there is a finite state s-gale for $s < 1$ that wins against it. So we need to show that if some s-gale wins against a selected subsequence, then there is some other s'-gale for (may be, different) $s' < 1$ that wins against the entire sequence. In terms of martingales: if some martingale wins exponentially fast playing against the selected (infinite) subsequence, then some other martingale wins exponentially fast against the entire sequence (may be, with different exponent).

In this language the idea of the proof is obvious. Assume that we have some strategy σ that plays against the subsequence. Then we can play against the entire sequence as follows. We make the trivial bets $(0.5 + 0.5)$ when we bet on the non-selected bits of the entire sequence, and use σ to bet on the selected bits. Since the selection rule is defined by a finite automaton, it is easy to see that the new strategy also has finite memory, and the capital will be the same as in the game of σ against the subsequence. The only problem is the rate: if the selected subsequence is very sparse, then exponential rate in the subsequence game is no more an exponential rate in the entire game. But the selected subsequence cannot be too rare, its density is separated from zero, as the following lemma says:

Lemma 17.1. *If the selected subsequence is infinite, then it has a positive density, i.e., the* lim inf *of the density of the selected terms is positive.*

Proof of the lemma. Consider a deterministic finite automaton that recognizes the set S. We denote this automaton by the same letter S. Let X be the set of states of S that appear infinitely many times when S is applied to α. Starting from some moment, the automaton is in X, and X is strongly connected (when speaking about strong connectivity, we ignore the labeling of the transition edges). Let us show that *vertices in* X *have no outgoing edges that leave* X. If these edges exist, let us construct a string u that forces S to leave X when started from any vertex of X. This will lead to a contradiction: a normal sequence has infinitely many occurrences of u, and one of them appears when S already is in X.

How to construct this u? Take some $q \in X$ and construct a string u_1 that forces S to leave X when started from q. Such a string u_1 exists, since X is strongly connected, so we can bring S to any vertex and then use the letter that forces S to leave X. Now consider some other vertex $q' \in X$. It may happen that u_1 already forces S to leave X when started from q'. If not and S remains in X (being in some vertex v), we can find some string u_2 that forces S out of X when started at v. Then the string u_1u_2 forces S to leave X when started in any of the vertices q, q'. Then we consider some other vertex q'' and append (if needed) some string u_3 that forces S to leave X when started at q, q' or q'' (in the same way). Doing this for all vertices of X, we get the desired u (and the desired contradiction).[9]

So X has no outgoing edges (and therefore is a strongly connected component of S's graph). Now the same argument shows that there exists a string u that forces S to visit all vertices of X when started from any vertex in X. This string u appears with positive density in α. So either the selected subsequence is finite (if X has no accepting vertices) or the selected subsequence has positive density (since every occurrence of u means that at least one term is selected when S visits the accepting vertex). Lemma 17.1 is proven. □

This finishes the proof of Theorem 17. □

Now we return to the first claim of Theorem 15. We postponed the proof of the following result: *no finite state martingale wins against a normal sequence.* Now we are ready to prove it and even a slightly stronger statement [38]:

[9] One may use also a probabilistic argument: for every vertex $q \in X$ there is some string that forces S to leave X when started at q, so for a sufficiently long random input string the probability to remain all the time in X is very small. And if it is smaller that $1/|X|$, there is an input string that works for all $q \in X$.

Theorem 18 (Schnorr and Stimm)**.** *Assume that α is a normal sequence and m is a finite state martingale. Then either the values of m on the prefixes of α are constant, starting from some moment, or they decrease exponentially fast.*

Proof. First we use the same argument as in the proof of Lemma 17.1. Look at the states of the finite-state martingale and consider the set X of states that appear infinitely often. As we have seen, this set has no outgoing edges and every state has positive $\liminf$-density. Consider some state i from X. It has two outgoing edges. Theorem 17 guarantees that these two edges are used equally often (in the limit), since every subsequence selected by a finite automaton is normal (imagine that this state is used as the unique accepting state in the selection rule). Does the martingale makes a non-trivial bet in the state i? If it does not, for all i, i.e., if the martingale makes equal bets in all states from X, then the capital remains constant since the game stays in X starting from some moment. If there is some $i \in X$ where the bets are not equal (say, p and $1-p$ fractions of capital are used, where $p \neq 1/2$), then using each of the two edges once, we multiply the capital by $4p(1-p)$, and this number is less than 1. So we get a factor less than 1 for every $i \in X$ with non-equal bets, and get factor 1 for all $i \in X$ that have equal bets, so the capital decreases exponentially fast (recall the every state in X is visited with positive density). □

5.8 Multi-account gales and a more general notion of finite state measures

As we have said, the finite state s-gales in our sense are called 1-*account* s-gales in [16]. In the same paper a more general notion, called *finite state k-account s-gales*, is considered. They can be defined as non-negative linear combinations of 1-account finite state s-gales. The intuitive motivation is clear: the gambler splits her capital into k different "accounts" and for each account uses the finite-state strategy (but never transfers money between the accounts).

Obviously, for the dimension of individual sequences (the case we are considering) this does not change anything: to win against the sequence, a multi-account strategy should contain a winning sub-strategy for some account.

From the viewpoint of gambling strategy the notion of finite state martingale looks quite reasonable. However, if we consider just output distribution of random processes with finitely many states, there is a natural generalization. Assume that a finite set of states is given, and one of them is chosen as an initial state. Assume also that for every state there are several outgoing edges, each has some transition probability, and for every state the sum of the probabilities for all outgoing edges equals 1. This defines a random walk, and if we add for each edge a bit label (0 or 1), we get a probability distribution on the infinite bit sequences. (The difference with the previous definition is that now the state is not determined by the output string.) The output distribution of this type can then be used to define martingales and s-gales in the same way, giving a more general definition.

It is easy to see that k-account gales become a special case of this definition (the splitting of the money between accounts is replaced by a probabilistic choice on the first step). Also one can define the version of KA_R based on this more general definition (taking the maximum over all states as initial states), and this version can be also used to characterize the finite state dimension, as the following lemma implies.

Lemma 18.1. *The function* KA_R *in this general version is also a superadditive upper bound for a priori complexity.*

Proof. Indeed, the distribution obtained for every fixed initial state is a computable measure on the Cantor space, so for each initial state we get an upped bound for a priori complexity, and the minimal of these bounds is still an upper bound.

To show the superadditivity, we cannot anymore use the equality $\rho_i(uv) = \rho_i(u)\rho_j(v)$ since now the process can be in different states after output u. We need to replace $\rho_j(v)$ by a weighted sum of $\rho_k(v)$ for different k that can be the states after output u. But since we take the maximum of $\rho_k(v)$ for all k when defining KA_R, we still have the superadditivity property. □

6 Discussion

The connection between normality and finite-state computations was noticed long ago, as the title of [1] shows; see also [38] where normality was related to martingales arising from finite automata. This connection led to a characterization of normality as incompressibility (see [4] for a direct proof). On the other hand, it was also clear that the notion of Kolmogorov complexity is not directly practical since it considers arbitrary algorithms as decompressors, and this makes it non-computable. So restricted classes of decompressors are of interest, and finite-state computations are a natural candidate for such a class.

Shallit and Wang [39] suggested to consider, for a given string x, the minimal number of states in an automaton that accepts x but not other strings of the same length. Later Hyde and Kjos-Hanssen [20] considered a similar notion using nondeterministic automata. The intrinsic problem of this approach is that it is not naturally "calibrated" in the following sense: measuring the information in bits, we would like to have about 2^n objects of complexity at most n. A "calibrated" approach was suggested by Calude, Salomaa and Roblot [11]; we have already discussed their definition in Section 4.6.

The incompressibility notion used in [4] provides such a characterization for yet another approach to automatic complexity. It uses deterministic transducers applied to a sequence whose complexity is measured. Becher and Heiber require additionally that for every output string y and every final state s there is at most one input string that produces y and brings the automaton into the state s. Our approach is a refinement of this one: we consider non-deterministic automata without initial/final states and require only that decompressor is an $O(1)$-valued function. The proofs become simpler for two reasons: (1) we compare the automatic complexity and Kolmogorov complexity and use standard results about Kolmogorov complexity; (2) we explicitly state and prove the property $\mathrm{C}_R(xy) \geqslant \mathrm{C}_R(x) + \mathrm{C}_R(y)$ that is crucial for the proofs.

An interesting open question is to find out the relations between different automatic complexity notions. Is there any formal relation between automatic complexity as defined in Section 2 and notions of finite state a priori complexity as defined in Section 5.6 and 5.8? Does the generalization of the class of probabilistic processes (Section 5.8) change the class of the corresponding complexity functions? Note that all three notions (automatic complexity and two finite state a priori complexities) can be used to characterize normality since they are all superadditive and are upper bounds for Kolmogorov complexity with logarithmic precision. Still the results showing the different definitions of Kolmogorov complexity (using a priori probability and description modes) are close to each other, do not imply that the finite state versions of the same notions are also close to each other.

Acknowledgments

We are grateful to Veronica Becher, Olivier Carton and Paul Heiber for many discussions of their paper [3] and the relations between incompressibility and normality, and for the permission to use the observations made during these discussions in the current paper.

The initial version of this paper (that does not deal with finite state dimension and does not contain the complexity criterion for normality) was presented at Fundamentals of Computation Theory symposium in 2017 [43] and is available in `arxiv` as [42]. The results about finite state dimension and the complexity criterion were presented at Fundamentals of Computation Theory symposium in 2019 [21].

We are grateful to the colleagues in LIRMM (ESCAPE team) and Moscow (Kolmogorov seminar, Computer Science Department of the HSE), and to the anonymous reviewers of STACS and FCT conferences; many of their remarks are taken into account in the current version.

References

[1] V.N. Agafonov. Normal sequences and finite automata, *Doklady AN SSSR*, **179**, 255–256 (1968). See also the paper of V.N. Agafonov with the same name in the collection: *Problemy Kibernetiki* (Cybernetics problems). Volume 20. Moscow: Nauka, 1968, p. 123–129.

[2] V. Becher, Turing's normal numbers: towards randomness, *How the World Computes*, Proceedings of the Turing Centenary Conference and the 8th Conference in Computability in Europe, CiE2012, Cambridge, UK, June 18–23, 2012. Lecture notes in computer science, *7318*, 35–45. Springer-Verlag, 2012.

[3] V. Becher, O. Carton, P. Heiber, *Finite-state independence*, 12 November 2016, `https://arxiv.org/pdf/1611.03921.pdf`

[4] V. Becher, P. Heiber, Normal number and finite automata, *Theoretical Computer Science*, **477**, 109–116 (2013).

[5] J. Berstel, *Transductions and Context-Free Languages* Vieweg+Teubner Verlag, 1969. ISBN: 978-3-519-02340-1. (For a revised 2006–2009 version see the author's homepage, `http://www-igm.univ-mlv.fr/~berstel`)

[6] Abram S. Besicovitch, The asymptotic distribution of the numerals in the decimal representation of the squares of the natural numbers, *Mathematische Zeitschrift*, **39**(1), 146–156 (1935), `http://link.springer.com/10.1007/BF01201350`

[7] Laurent Bienvenu, Glenn Shafer, Alexander Shen, On the history of martingales in the study of randomness, *Electronic Journ@l for History of Probability and Statistics*, **5**(1), June 2009, 1–40, `http://www.jehps.net/juin2009/BienvenuShaferShen.pdf`

[8] Émile Borel (Paris), Les probabilités dénombrables et leur applications arithmétiques, *Rendiconti del Circolo Matematico di Palermo*, **27**(1), 247–271 (2009), `https://doi.org/10.1007/BF03019651`

[9] Chris Bourke, John M. Hitchcock, N.V. Vinodchandran, Entropy rates and finite-state dimension, *Theoretical Computer Science*, **349**(3), 392–406 (2005), `https://doi.org/10.1016/j.tcs.2005.09.040`

[10] Y. Bugeaud, *Distribution modulo one and Diophantine approximation*, Cambridge Tracts in Mathematics, 193, Cambridge University Press, 2012.

[11] C.S. Calude, K. Salomaa, T.K. Roblot, Finite state complexity, *Theoretical Computer Science*, **412**(41), 5668–5677 (2011).

[12] Cristian S. Calude, Ludwig Staiger, Frank Stephan, Finite state incompressible infinite sequences, *Information and Computation*, **247**, 23–36 (2016), `https://doi.org/10.1016/j.ic.2015.11.003`

[13] J.W.S. Cassels, On a paper of Niven and Zuckerman, *Pacific Journal of Mathematics*, **2**(4), 555–557, `http://msp.org/pjm/1952/2-4/p06.xhtml`

[14] D. Champernowne, The construction of decimals normal in the scale of ten, *Journal of the London Mathematical Society*, volume s1-8, issue 4 (October 1933; Received 19 April, read 27 April,1933), 254–260.

[15] Arthur H. Copeland, Paul Erdös, Note on normal numbers, *Bulletin of the American Mathematical Society*, **52**(10), 857–860 (1946), `https://projecteuclid.org/euclid.bams/1183509721`.

[16] Jack J. Dai, James I. Lathrop, Jack H. Lutz, Elvira Mayordomo, Finite-state dimension, *Theoretical Computer Science*, **310**, issues 1–3, p. 1–33 (2004), `https://doi.org/10.1016/S0304-3975(03)00244-5`.

[17] David Doty, Jack H. Lutz, Satyadev Nandakumar, Finite-state dimension and real arithmetic, *Information and Computation*, **205**(11), 1640–1651 (2007), `https://doi.org/10.1016/j.ic.2007.05.003`

[18] R.G. Downey, D.R. Hirschfeldt, *Algorithmic randomness and complexity*, Springer, 2010, ISBN 978-0-387-68441-3, xxviii+855 p.

[19] H. Daverport, P. Erdös, Note on Normal Decimals, *Canadian Journal of Mathematics*, **4**, 58–63 (1952)

[20] K.K. Hyde, B. Kjos-Hanssen, Nondeterministic complexity of overlap-free and almost square-free words, *The Electronic Journal of Combinatorics*, **22**:3, 2015.

[21] Alexander Kozachinsky, Alexander Shen, Two Characterizations for Finite-State Dimension, *Fundamentals of Computation Theory*, 22nd International Symposium, FCT 2019 Copenhagen, Denmark, August 12–14, 2019, Springer, Lecture Notes in Computer Science, v. 11651 (2019), 80–94.

[22] L. Kuipers, H. Niederreiter, *Uniform distribution of sequences*. John Wiley & Sons, 1974.

[23] M. Li, P. Vitányi, *An Introduction to Kolmogorov complexity and its applications*, 3rd ed., Springer, 2008 (1 ed., 1993; 2 ed., 1997), 792 pp. ISBN 978-0-387-49820-1.

[24] Jack H. Lutz, Dimension in complexity classes. *SIAM Journal on Computing*, **32**(5), 1236–1259, `https://doi.org/10.1137/S0097539701417723`. Preliminary version appeared in *Proc. 15th IEEE Conference on Computational Complexity* (CCC 2000), p. 158– 169.

[25] Jack H. Lutz, The dimension of individual strings and sequences, *Information and Computation*, **187**(1), 49–79 (2003), `https://doi.org/10.1016/S0890-5401(03)00187-1`

[26] Elvira Mayordomo, A Kolmogorov complexity characterization of constructive Hausdorff dimension, *Information Processing Letters*, **84**, 1–3 (2002), `https://doi.org/10.1016/S0020-0190(02)00343-5`

[27] John E. Maxfield, A short proof of Pillai's theorem of normal numbers, *Pacific Journal of Mathematics*, **2**(1), 23–24 (1952), `http://msp.org/pjm/1952/2-1/p04.xhtml`

[28] A. Nies, *Computability and randomness*, Oxford Logic Guides, Oxford University Press, 2009, ISBN 978-0199652600, 435 p.

[29] Ivan Niven, *Irrational Numbers*, The Mathematical Association of America, John Wiley and Sons, Inc., 1956, vii+164 p. (Chapter VIII. Normal numbers, p. 94–116)

[30] Ivan Niven and H.S. Zuckerman, On the definition of normal numbers, *Pacific Journal of Mathematics*, **1**(1), 103–109 (1951), `https://projecteuclid.org/euclid.pjm/1102613156`

[31] I.I. Piatetski-Shapiro, On the laws of distribution of the fractional parts of an exponential function [Russian: И. И. Пятецкий-Шапиро, О законах распределения дробных долей показательной функции], Izvestia Akademii Nauk SSSR, Ser. Matem., **15**(1), 47–52 (1951). In Russian.

[32] I.I. Piatetski-Shapiro, On the distribution of the fractional parts of an exponential function [Russian: И. И. Пятецкий-Шапиро, О распределении дробных долей показательной функции], *Uchenye Zapiski Moskovskogo Gosudarstvennogo Pedagogicheskogo Instituta*, **108**(2), 317–322 (1957). In Russian.

[33] S.S. Pillai, On normal numbers, *Proceedings of the Indian Academy of Sciences*, Section A, **10**(1), 13–15 (1939), `https://www.ias.ac.in/article/fulltext/seca/010/01/0013-0015`

[34] S.S. Pillai, On normal numbers, *Proceedings of the Indian Academy of Sciences*, Section A, **12**(2), 179–184 (August 1940), `https://www.ias.ac.in/article/fulltext/seca/012/02/0179-0184`

[35] A.G. Postnikov, On the question of distribution of fractional parts of the exponential function [Russian: А. Г. Постников, К вопросу о распределении дробных долей показательной функции], Doklady AN SSSR, **LXXXVI** (3), 473–476 (1952). In Russian.

[36] A.G. Postnikov, I.I. Piatetski-Shapiro, Bernouilli-normal sequences of symbols [Russian: Нормальные по Бернулли последовательности знаков], *Izvestia AN SSSR, Seriya matematicheskaya*, **21**(4), 501–514 (1957), `http://mi.mathnet.ru/izv4030`. In Russian.

[37] C. Schnorr, A unified approach to the definition of random sequences, *Mathematical Systems Theory* (now *Theory of Computing Systems*), **5**(3), 246–258 (1971), `https://doi.org/10.1007/BF01694181`

[38] C. Schnorr, H. Stimm, Endliche Automaten und Zufallsfolgen, *Acta Informatica*, **1**(4), 345–39 (1972).

[39] J. Shallit, M.-W. Wang, Automatic complexity of strings, *Journal of Automata, Languages and Combinatorics*, **6**:4. 537-554 (April 2001)

[40] A. Shen, Around Kolmogorov complexity: basic notions and results. *Measures of Complexity. Festschrift for Alexey Chervonenkis*, edited by V. Vovk, H. Papadoupoulos, A. Gammerman, Springer, 2015, p. 75–116, see also `http://arxiv.org/abs/1504.04955` (2015)

[41] A. Shen, V.A. Uspensky, N. Vereshchagin, *Kolmogorov complexity and algorithmic randomness*, Moscow, MCCME, 2013 (In Russian). English version published by AMS, see `http://www.ams.org/publications/authors/books/postpub/surv-220` and `www.lirmm.fr/~ashen/kolmbook-eng-scan.pdf`.

[42] A. Shen, *Automatic Kolmogorov Complexity and Normality Revisited*, `https://arxiv.org/abs/1701.09060v1`, preliminary version of the current text.

[43] A. Shen, Automatic Kolmogorov Complexity and Normality Revisited, *Fundamentals of Computation Theory, 2017, Proceedings*, Lecture Notes in Computer Science, vol. 10472, 418–430 (2017)

[44] V.A. Uspensky, A. Shen, Relations between varieties of Kolmogorov complexities, *Mathematical Systems Theory*, **29**, 271–292 (1996).

[45] Jean Ville, *Étude critique de la notion de collectif*, Gauthier-Villars, Paris, 1939, `http://www.numdam.org/issue/THESE_1939__218__1_0.pdf`. In French.

[46] D.D. Wall, *Normal numbers*, Thesis, University of California, 1949.

[47] A. Weber, On the valuedness of finite transducers, *Acta Informatica*, **27**(8), 749–780 (1990).

[48] Hermann Weyl in Zürich, Über dir Gleichverteilung von Zahlen mod. Eins, *Mathematische Annalen*, **77**(3), 313–352 (1916), `http://link.springer.com/10.1007/BF01475864`