Automatic Kolmogorov complexity and normality revisited

Alexander Shen*

LIRMM CNRS & University of Montpellier.
On leave from IITP RAS, Moscow, Russia
alexander.shen@lirmm.fr

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 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 (the 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 [13], Calude–Salomaa–Roblot [6]) of description complexity related to finite automata are discussed (see the last section).
As a byproduct, this approach provides simple proofs of classical results about normality (equivalence of definitions with aligned occurences and all occurencies, Wall's theorem saying that a normal number remains normal when multiplied by a rational number, and Agafonov's result saying that normality is preserved by automatic selection rules).

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; a historical account can be found in [4]).

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 [7, 11] for the detailed exposition, [15] for a textbook and [14] for a short survey). The most popular definition 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

* Supported by ANR-15-CE40-0016-01 RaCAF grant

if an only if prefixes of α are incompressible (do not have short descriptions). See again [7, 11, 15, 14] for the 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 realized 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 this characterization does not fit well the general framework of Kolmogorov complexity (finite automata are considered 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. In Section 3 we recall the notion of normal sequence. Then in Section 4 we provide a characterization of normal sequences in terms of automatic Kolmogorov complexity. In Section 5 we show how this characterization can be used to give simple proofs for classical results about normality, including Wall's theorem (normal numbers remain normal when multiplied by a rational factor). In a similar way one can prove Agafonov's result [1], but this part is omitted due to the space restrictions (see the `arxiv` version of this paper or the Appendix). Finally in Section 7 we compare our definition of automatic complexity with other similar notions.

2 Automatic Kolmogorov complexity

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 "program", or "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 kinds of "descriptions" we consider, but most versions of Kolmogorov complexity can be described according to this scheme [16]:

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 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 Kolmogorov–Solomonoff optimality theorem says that there exist an optimal D in this class that makes C_D minimal (up to $O(1)$ additive term). (We assume that the reader is familiar with basic properties of Kolmogorov complexity, see, e.g., [9, 15]; for a short introduction see also [14].)

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 objects.

To define our class of descriptions, 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 point), and for each path concatenate all the first and all the second components of the pairs; 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".

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. Now there is no optimal mode D that makes C_D minimal (see below Theorem 1). So, stating some properties of complexity, we need to mention D explicitly and say something like "for every automatic description mode D there exist 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.)

Let us first mention 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 $\{(p, x0) : (p, x) \in D\}$ 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 is obvious in all three cases. Let us prove the first (by a standard argument 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 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 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 on $K \times L$ guarantees that $q = u$ and 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 inbetween 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.

Now we are ready to prove the following simple result stating 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 $\mathrm{C}_R(x) \le |x|$ for all strings x.*

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

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

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

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

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

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

(h) *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 $\mathrm{C}_R(x) \le \mathrm{C}_{R'}(x) + c$ for all strings x.)*

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

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

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

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

(m) *If we allow a bigger alphabet B instead of $\mathbb{B}$ 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 $O(1)$-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, 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, is only $O(1)$ bits longer than x.

(e) 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 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 $(k-1)c \le l \le (k+1)c$.

(f) 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 that c.

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

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

(i) 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 input symbols used while producing x'.

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

(k) 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-,…, Nth layers). The outgoing edges from the vertices of ith 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 ith layer as before). The edges from Nth layer are of two types: for each vertex x there is an edge with label $(0, \varepsilon)$ that goes to the same vertex in 0th layer, and edges with labels $(1, \varepsilon)$ that connect each vertex of Nth 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 in $O(1)$-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.

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

(m) Take the composition of a given description mode R with a mode that provides block coding of inputs. Note that block coding can be implemented by an automaton (first read the code, the output the encoded word). 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 coding with large block size.

Remark 1. Not all these results are used in the sequel; we provide them for comparison with the properties of standard Kolmogorov complexity.

3 Normal sequences and numbers

Consider an infinite bit sequence $\alpha = a_0 a_1 a_2 \ldots$ and some integer $k \geq 1$. Split the sequence α into k-bit blocks: $\alpha = A_0 A_1 \ldots$. For every k-bit string r consider the limit frequence 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. *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 this definition that considers all occurences of some string r in α, not only aligned ones (whose starting point is a multiple of k). In this 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 (see, e.g., [10, Chapter 1, Section 8]) says that this is an equivalent notion, and we give below 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* [5]

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

(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 probability close to 2^{-k}, since each of 2^{N-1} strings (after the leading 1 for N-bit numbers in the Champernowne sequence) appears exactly once. The deviation is caused by the leading digits 1 and also by the boundaries between consequtive 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 (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 subblocks, and the distance can be made small compared to the total length of the prefix. (End of the proof sketch.)

The definition of normality can be given for arbitrary alphabet (instead of the binary one), so we get also the notion of *b-normality* of a real number for every base $b \geq 2$. It is known that the normality for different bases is not equivalent (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. For every base b, almost all reals are b-normal (the non-normal numbers have Lebesgue measure 0); this is guaranteed by the Strong Law of Large Numbers. Therefore the numbers that are not absolutely random form a null set (a countable union of null sets for all b). The constructive version of this argument shows that there exist computable absolutely normal numbers, the result that 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 its length.

4 Normality and incompressibility

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

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

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 the compactness argument, 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, 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_1A_2 whose values are $2k$-bit blocks (and A_1 and A_2 are their first and second halves of length k). The variables A_1 and A_2 may be dependent, and the distribution may differ from the initial distribution A for k-bit blocks. Still we know that A is the average of A_1 and A_2 (since A is computed for all blocks, and A_1 [resp. A_2] corresponds to odd [resp. even] blocks). The convexity argument (the function $p \mapsto -p \log p$ used in the definition of entropy has negative second derivative) shows that $H(A) \geq [H(A_1) + H(A_2)]/2$. Then

$$H(A_1A_2) \leq H(A_1) + H(A_2) \leq 2H(A),$$

so A_1A_2 has twice bigger difference between entropy and length. 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 every 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

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

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

$$\liminf_{n\to\infty} \frac{\mathrm{C}_R(A_0A_1\ldots A_{n-1})}{nk} \geq 1 - \frac{O(1)}{k},$$

where the constant in $O(1)$ does not depend on k. This will be enough: note that (i) we may consider only prefixes whose length is a multiple of k, because 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 the bound for some fixed k. Recall that

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

and that $\mathrm{C}(x) \leq \mathrm{C}_R(x) + O(1)$ 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 among A_0, A_1,..., A_{n-1}. It remains to note that 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.

A small modification of this proof adapts it to the non-aligned definition of normality. Let α be a sequence that is not normal in the non-aligned version. This means that for some k all k-bit blocks (non-aligned) do not have a 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 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 other sense (with aligned blocks). 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 at most by $O(1)$.

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.

Indeed, let us consider blocks of length k whose C_R-complexity is smaller than $k - d$. Their Kolmogorov complexity is then smaller than $k - d + O(1)$, and the fraction of these blocks (among all k-bit strings) is at most $2^{-d+O(1)}$. So their frequency among aligned blocks is at most $2^{-d+O(1)} \cdot k$. For all other blocks R-compression saves at most d bits, and for 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

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

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

So we get the following corollary:

Corollary 1. *The aligned and non-aligned definitions of normality are equivalent.*

Note also that we proved that adding/deleting a finite prefix does not change the compressibility, and, therefore, normality. (For a non-aligned version of normality definition it is obvious anyway, but for the aligned version it is not so easy to see directly.)

5 Wall's theorem

Now we obtain a known result about normal numbers (Wall's theorem) as a easy corollary. Recall that a real number is normal if its binary expansion is normal. Recall that we agreed to ignore the integer part (since it has only finitely many digits, adding it as a prefix would not matter anyway).

Theorem 3 (Wall [17]). *If p and q are rational numbers and α is normal, then $\alpha p + q$ is normal.*

Proof. It in enough to show that multiplication and division by an integer c preserve normality (note that adding an integer does not change it by definition, since the integer part is ignored).

This fact follows from the incompressibility characterization (Theorem 2), the non-increase of complexity under automatic $O(1)$ mappings (Theorem 1, (l)) and the following lemma:

Lemma 1. *Let c be an integer. Consider the relation R_c that consists of pairs of strings x and y such that x and y have the same length and can be prefixes of the binary expansions of the fractional parts of γ and $c\gamma$ for some real γ. This relation, as well as its inverse, is contained in an automatic description mode.*

Assuming the Lemma, we conclude that the prefix of γ and $c\gamma$ have the same automatic complexity. More precisely, for every automatic description mode R there exists another automatic description mode R' such that $\mathrm{C}_{R'}(y) \leq \mathrm{C}_R(x)$ if x and y are prefixes of γ and $c\gamma$ respectively. So if γ is compressible, then $c\gamma$ is also compressible. The same is true if we consider the inverse relation; if γ is compressible, then γ/c is also compressible.

It remains to prove the lemma. Indeed, school division algorithm can be represented by an automaton; the integer part can be different, but this creates $O(1)$ different possible remainders. We have to take care about two representations of the same number (note that while dividing $0.29999\ldots$ by 3, we obtain only $0.09999\ldots$, not $0,10000\ldots$), but at most two representations are possible and the relation between them is automatic, so we still get an automatic description mode.

6 Pairs as descriptions and Agafonov's theorem

The incompressibility criterion for normality can also be used for an easy proof of Agafonov's theorem from [1]. This result says that an automatic selection rule (a term a_n of a sequence is selected or not depending on whether $a_0 \ldots a_{n-1}$ is accepted by a finite automaton), being applied to a normal sequence, selects either finite or normal sequence.

The idea of the proof: a sequence can be split into two: the selected subsequence and the rest. The selection process guarantees that the original sequence can be reconstructed from these two subsequences. If one of them (the selected one) is compressible, then this compression can be used to compress the prefixes of the original sequence (the unselected part is given as is, but the selected part is compressed).

There are two technical points needed to implement this plan: first, one should prove that the selected sequence has positive density (using the normality of the original sequence); second, one should generalize the notion of automatic complexity by using pairs as descriptions.

Due to the lack of space, the details of this argument are moved to Appendix A (see also the arxiv version)

7 Discussion

The connection between normality and finite-state computations was realized long ago, as the title of [1] shows; see also [12] 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 [13] suggested to consider, for a given string x, the minimal number of states of an automaton that accepts x but not other strings of the same length. Later Hyde and Kjos-Hanssen [8] considered a similar notion using nondeterministic automata. The intrinsic problem of this approach is that it is not naturally "calibrated" in the following sence: if we intend to measure the information in bits, it is desirable to have about 2^n objects of complexity at most n.

Another (and "calibrated") approach was suggested by Calude, Salomaa and Roblot [6]: in their definition a deterministic transducer maps a description string to the string to be described, and the complexity of y is measured as combination of the sizes of a transducer and an input string that produce y (minimum over all transducers and all input strings producing y is taken). The size of the transducer is measured via some encoding, so the complexity function depends on the choice of this encoding. The open question posed in [6, Section 6] asks whether this notion of complexity can be used to characterize normality.

The incompressibility notion used in [4] provides such a characterization for a different definition. They consider deterministic transducers and require additionally that for every output string y and every final state s there exists at most one input string that produces y and brings the automaton in state s. Our approach is essentially a simplification and refinement of this one: we observe that it fits the scheme for Kolmogorov complexity and has nice properties if we consider non-deterministic automata without initial states and require only that decompressor is a $O(1)$-valued function. The proofs of the normality criterion and other results then become simpler, mainly for two reasons: (1) we use the comparison of automatic Kolmogorov complexity and plain Kolmogorov complexity and apply standard results about Kolmogorov complexity; (2) we explicitly state and prove the property $\mathrm{C}_R(xy) \geq \mathrm{C}_R(x) + \mathrm{C}_R(y)$ that makes automatic complexity different, and use it in the proof.

Acknowledgements. I am 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. I am also grateful to my colleagues in LIRMM (ESCAPE team) and Moscow (Kolmogorov seminar, Computer Science Department of the HSE). I am thankful to the anonymous referees of an early version of this paper submitted to ICALP (and rejected; see the reviews in Appendix B).

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. 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.
6. C.S. Calude, K. Salomaa, T.K. Roblot, Finite state complexity, *Theoretical Computer Science*, **412**, 5668–5677 (2011).
7. R.G. Downey, D.R. Hirschfeldt, *Algorithimic randomness and complexity*, Springer, 2010, ISBN 978-0-387-68441-3, xxviii+855 p.
8. K.K. Hyde, B. Kjos-Hanssen, Nondeterministic complexity of overlap-free and almost square-free words, *The Electronic Journal of Combinatorics*, **22**:3, #P3.22, 2015.
9. 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.
10. L. Kuipers, H. Niederreiter, *Uniform distribution of sequences.* John Wiley & Sons, 1974.
11. A. Nies, *Computability and randomness*, Oxford Logic Guides, Oxford University Press, 2009, ISBN 978-0199652600, 435 p.
12. C. Schnorr, H. Stimm, Endliche Automaten und Zufallsfolgen, *Acta Informatica*, **1**(4), 345–39 (1972).
13. J. Shallit, M.-W. Wang, Automatic complexity of strings, *Journal of Automata, Languages and Combinatorics*, **6**:4. 537-554 (April 2001)
14. 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)
15. A. Shen, V.A. Uspensky, N. Vereshchagin, *Kolmogorov complexity and algorithmic randomness*, Moscow, MCCME, 2013 (In Russian). English version accepted for publication by AMS, see www.lirmm.fr/ ashen/kolmbook-eng.pdf
16. V.A. Uspensky, A. Shen, Relations between varieties of Kolmogorov complexities, *Mathematical Systems Theory*, **29**, 271–292 (1996).
17. D.D. Wall, *Normal numbers*, Thesis, University of California, 1949.

8 Appendix A: Agafonov's theorem

In this section we derive another classical result about normal numbers, Agafonov's theorem [1], from the incompressibility characterization. (However, we will need to modify this characterization, see below).

Agafonov's result is motivated by the von Mises' approach to randomness (see, e.g., [15, Chapter 9] for the historic account). As von Mises mentioned, a "random sequence" (he used the German term "Kollektiv") should remain random after using a reasonable selection rule. More precisely, assume that there is some set R of binary strings, we observe a binary sequence $\alpha = a_0a_1a_2\ldots$ and select terms a_n such that $a_0a_1\ldots a_{n-1} \in R$ (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 4 (Agafonov). *Let $\alpha = a_0a_1a_2\ldots$ be a normal sequence. Let R be a regular (=recognizable by a finite automaton) set of binary strings. Consider a subsequence σ made of terms a_n such that $a_0a_1\ldots a_{n-1} \in R$ (in the same order as in the original sequence). Then σ is normal or finite.*

Proof. This proof adapts the arguments from [4] to our definition of compressibility. Using the incompressibility criterion, we need to prove that if the sequence σ is compressible, then α is compressible, too. The idea is simple: the selection rule splits α into two subsequences: the selected terms (σ) and the rest (the non-selected terms). We do not know anything about the second subsequence, but we assume that σ is compressible, and want to prove that the entire sequence α is compressible.

The key observation: knowing both subsequences (and the selection rule R, of course), we can reconstruct the original sequence. Indeed, we know whether a_0 is selected or not: it depends on whether the empty string belongs to R or not. So we know where we should look for the first term when reconstructing α from its two parts. Knowing a_0, we can check whether one-letter word a_0 belongs to R or not. So we know whether a_1 is selected, so we know from which subsequence it should be taken, etc.

So, we know that our sequence can be reconstructed from two its parts, and one part is compressible. Then the entire sequence is compressible: a compressed description consist of a compressed description of a compressible subsequence, and the trivial description of the other one (for which we do not know whether it is compressible or not). To make this argument precise, we need two things:

- prove that the selected subsequence has positive density (otherwise its compression gives only a negligible improvement);
- modify the definition of complexity and the criterion of compressibility allowing pairs as descriptions.

We start with the first part.

Lemma 2. *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 (Proof of the lemma). Consider a deterministic finite automaton that recognizes the set R. We denote this automaton by the same letter R. Let S be the set of states of R that appear infinitely many times when R is applied to α. Starting from some time, the automaton is in S, and S is strongly connected (when speaking about strong connectivity, we ignore the labeling of the transition edges). Let us show that *vertices in S have no outgoing edges that leave S*. If these edges exist, let us construct a string u that forces R to leave S when started from any vertex of S. This will lead to a contradiction: a normal sequence has infinitely many occurences of u, and one of them appears when R already is in S. How to construct this u? Take some $s \in S$ and construct a string u_1 that forces R to leave S when started from s. Such a string u_1 exists, since S is strongly connected, so we can bring R to any vertex and then use the letter that forces R to leave S. Now consider some other vertex $s' \in S$. It may happen that u_1 already forces R to leave S when started from s'. If not and R remains in S (being in some vertex v), we can find some string u_2 that forces R out of S when started at v. Then the string u_1u_2 forces R to leave S when started in any of the vertices s, s'. Then we consider some other vertex s'' and append (if needed) some string u_3 that forces R to leave S when started at s, s' or s'' (in the same way). Doing this for all vertices of S, we get the desired u (and the desired contradiction).

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

To finish the proof, we need to modify the notion of a description mode and consider pairs as descriptions. Let A, B, C be three alphabets. We define the notion of automatic ternary relation $R \subset A^* \times B^* \times C^*$ in the same way as for binary relations: now the edge labels are triples (a, b, c), where each of the letters (or even all three) can be replaced by ε-symbol. This relation can be considered as multivalued function of type $A \times B \to C$. If it is $O(1)$-valued, we call it *pair description mode*, and a pair (u, v) is called a *description of w* if $(u, v, w) \in R$. We assume, as before, that all the alphabets are binary ($A = B = C = \mathbb{B}$), and the length of description is measured as the sum of lengths:

$$\mathrm{C}_R(w) = \min\{|u| + |v| \colon (u, v, w) \in R\}.$$

The automatic description modes are special cases of pair description modes (if we use only one component of the pair as a description, and the other one is empty), but these generalization may lead to smaller complexity function. (It would be interesting to find out how much the decrease could be.) Still they have the properties we need:

Proposition 2. **(a)** *For every pair description mode*

$$\mathrm{C}(x) \le \mathrm{C}_R(x) + c \log \mathrm{C}_R(x) + c$$

for some c and all x, where $\mathrm{C}(x)$ stands for the Kolmogorov complexity of x.

(b) *If R is a pair description mode and $\alpha = a_0 a_1 \ldots$ is a normal sequence, then*

$$\lim \frac{\mathrm{C}_R(a_0 a_1 \ldots a_{n-1})}{n} = 1.$$

(c) *If a ternary relation $R(u, v, w)$ is a pair description mode and a binary relation $Q(u', u)$ is an automatic description mode, then their joint*

$$R'(u', v, w) = \exists u\, [Q(u', u) \text{ and } R(u, v, w)]$$

is a pair description mode.

(d) *Let R be a regular set of binary strings (recognized by a finite automaton) used as a selection rule. Then the relation*

$$\{(u, v, w)\colon$$
$$u \text{ and } v \text{ are strings of selected and non-selected bits when } R \text{ is applied to } w\}$$

is a pair description mode.

Proof (Proof of the Proposition). (a) Fix some pair description mode R. If $(u, v, w) \in R$, the string w is determined by the pair (u, v) and the ordinal number of x among the outputs of $O(1)$-valued function for input (u, v). So the Kolmogorov complexity of w exceed the Kolmogorov complexity of a pair (u, v) at most by $O(1)$, and complexity of a pair (u, v) is bounded by $l + O(\log l)$ where l is the total length of u and v.

(b) As before, we cut α into blocks of some large length k. Now the R-complexity of a block can be smaller than its Kolmogorov complexity, and the decrease can be $O(\log k)$, but this does not matter: for large k this change is negligible compared to k.

(c) The joint of two automatic relations is automatic, for the same reasons; the corresponding function is $O(1)$-valued since for each values of u' we have $O(1)$ different values of u, and each of them leads to $O(1)$ values of w (for a fixed v).

(d) The process of splitting w into u and v is automatic for obvious reasons. The notion of automatic relation does not distinguish between input and output, so this ternary relation is automatic. As we have discussed, for a given u and v there exists at most one w that can be obtained by merging u and v; to determine whether we take the next letter from u or v, we check whether the string of symbols already added to w belongs to the selection rule R. (Now the initial state is not fixed anymore, still we can at most $O(1)$ values for given u and v.)

Now we can finish the proof of Agafonov's theorem. Assume that some selection rule R is applied to a normal sequence α and selects some its subsequence σ that is not normal. After finitely many steps R splits a prefix a of α into sequence of selected terms s (it is a prefix of σ) and the sequence u of non-selected terms. Then (s, u, a) belongs to the pair description mode from part (d) of the proposition; let us denote it by U. Now recall that σ is not normal, so Theorem 2 says that there exists some description mode Q such that $\mathrm{C}_Q(s) < (1-\varepsilon)|s|$ for some $\varepsilon > 0$ and for infinitely many prefixes s of σ. Then use part (c) of the proposition and consider the joint J of Q and U. The

$$\mathrm{C}_J(a) \le \mathrm{C}_Q(s) + |u| \le (1-\varepsilon)|s| + |u| \le |s| + |u| - \varepsilon|s|.$$

for infinitely many prefixes a of α that correspond to compressible prefixes s of σ. Lemma guarantees that $\varepsilon|s|$ is $\Omega(|a|)$, so we use Theorem 2 in the other direction and get a contradiction with the normality of a.

9 Appendix B: disclosure/reviews

This paper was submitted to ICALP 2017 and rejected. For the convenience of the program committee of FCT the reviews (in full) are attached; note that ICALP has another TEX-style, so the references to line/page numbers are not correct (and anyway I've corrected the errors pointed out by the referees and made many of the improvements suggested), still the program committee may be interested in the reasons for rejection etc.

```
----------------------- REVIEW 1 ---------------------
PAPER: 15
TITLE: Automatic Kolmogorov complexity and normality revisited
AUTHORS: Alexander Shen


----------- Overall evaluation -----------
The paper considers the problem of describing factors of given
length that appear with same frequency in an infinite sequence.
The work is concerned with the Kolmogorov complexity of the
problem. While previous incompressibility result does not follow
the standard Kolmogorov complexity, this work revisits the
problem and presents results in the traditional Kolmogorov
complexity model.


I am not an expert related to Kolmogorov's complexity, and the
result presented might be interesting to the experts. However,
the manuscript is not written in an accessible way to a broader
audience. I mention some of the key problems below:

1. The paper does not have an Introduction. In a broad
conference like ICALP I would expect a paper to explain the
problem and its significance to a wide community. This is
```

completely lacking.

2. The significance of the normality problem is not explained. Why the problem is relevant and significant and where it is used is not clarified.

3. While the technical treatment might be adequate for the experts, the key ideas and intuition are not highlighted.

As a non-expert, I find the manuscript not suitable to a broad and top conference like ICALP.

----------------------- REVIEW 2 ---------------------

PAPER: 15
TITLE: Automatic Kolmogorov complexity and normality revisited
AUTHORS: Alexander Shen

----------- Overall evaluation -----------

The main aim of the paper is to give a reformulation in the style of Kolomogorov complexity of a characterization of "normal" sequences by incompressibility by finite state machines.

Let us recall that a infinite sequence of symbols is said to be normal if all possible blocks of the same length occur with the same frequency in it (if the alphabet has size k, all symbols occurs with frequency 1/k, all blocks of length 2 occur with frequency 1/k^2 and so on ...)

A classical result mainly due to Schnorr and Stimm states that a sequence is normal if and only if it cannot be compressed by (one-to-one) finite state machine, aka transducer. This statement has the flavour of the incompressibility by Turing machines of Martin-Lf random sequences. However, as pointed by the author, the two statements are not alike. In the former one, finite state machines are used to compress while in the latter one, Turing machines are used to de-compress in Kolmogorov complexity.

To fill the gap, the author develops a theory of "automatic Kolmogorov complexity" which allows him to rephrase Schnorr and Stimm's result. It is always good to have a different view point and this paper is interesting for that reason. However, the new approach has the following weaknesses: + The new approach does not provide any new result + The new approach does not simplify the proof of some results.

- The proof of the equivalence between normality and incompressibility is no more than one page. See [4] cited in the abstract of the paper.

- Once this result is known, all statements mentioned in the paper like Wall's theorem and Agafanov's result have short proofs.

For these reasons, the paper does nor reach the standard of ICALP although it deserves to be published.

----------------------- REVIEW 3 ---------------------

PAPER: 15
TITLE: Automatic Kolmogorov complexity and normality revisited
AUTHORS: Alexander Shen

----------- Overall evaluation -----------

The paper "Automatic Kolmogorov complexity and normality revisited" considers a compression with a finite state automata based computation model and Kolmogorov complexity with regards to this computation model.

The used computation model is a finite-state transducer with every state to be an initial state and every state to be a final state. Furthermore, there has to be a constant c for each considered transducer such that each input string has at most c many possible output strings.

For a fixed transducer D of that kind, C_D(x) is a complexity measure for the string x, the minimal length of input strings to allow D to produce the output string x. This C_D is an automatic variant of the Kolmogorov complexity. It is shown that C_D is an upper bound for the Kolmogorov complexity: The input for the description as well as a number to denote which of the possible outputs is x unambiguously define the string x. It is also mentioned implicit that the Kolmogorov complexity can be much smaller than the automatic Kolmogorov complexity: The Champernowne number which has a very small Kolmogorov complexity has a very big automatic Kolmogorov complexity.

Basic properties of the automatic Kolmogorov complexity are then proven. These are then used to analyze properties of normal sequences. Wall's theorem and Agafonov are then reproven using automatic Kolmogorov complexity.

The paper is mostly well-written, most proofs are easy to follow. While not ground-breaking the paper still delivers interesting progress: A simple model with good analysis of its properties. Theorem 5 for example covers all basic properties of automatic Kolmogorov complexity, the proof is well-divided and is easy to check.

The herein defined automatic Kolmogorov complexity is for sure not the only way to define it, but given the properties that come out of this definition, it seems to be a good choice to define it that way.

On the downside the paper has some weaknesses, mostly in the presentation: The paper starts without a summarizing introduction. The same thing happens in the abstract: automatic Kolmogorov complexity is the main result and the normality results are "a byproduct" (explicit stated as such in the abstract). Nevertheless, the abstract starts with normality making automatic Kolmogorov complexity appear as side result. Also the automatic selection procedure in Theorem 11 is hard to understand from this paper alone.

A mistake seems to have slipped into Theorem 7: It states that the limit = 1, but the proof is only performed for >= 1. Indeed, I think there are transducers to produce a bigger value in the limit, for example 2 with the following:

```
D=
o --(0,0)(1,1)--> o --(1,epsilon)--\
^                                  |
\----------------------------------/
```

C_D(x) is 2*|x|-1 if |x|>0, hence this limes is 2.

The last downside I want to mention is that there is no procedure given to check whether some transducer is even in this normal form (that it is O(1)-valued). I think this is not so complicated, but regardless whether it is simple or not: I should be at least mentioned.

All in all the paper is well-written, the results interesting and the downsides are not that big and also fixable, so I suggest acceptance.

Minor remarks:

p1l-13...p2l6: This part can be split in three blocks: "Details of definition" "Definition itself" "comparison to Kolmogorov complexity" I think it would make much more sense to reverse the order of these three blocks and to include some intuition in the then-first block "comparison to Kolmogorov complexity" what is planed to do in the paper, e.g. "upper bound to Kolmogorov complexity by a simpler comutation model, a finite state transducer"

Definition 1,2,3: O(1)-valued functions are introduced in Definition 3; this aspect seemes to be missing in Definition 1 and 2. While it is clear when reading the remaining paper that this is meant to be part of the C_D-definition, it is not there.

p2l-7: x0 confused me when I read it the first time, it took a while to realize that 0 as element of the binary alphabet is meant. As it is on top of the same page it should be ok anyway, but if you happen to see a way to improve that, do it

p5l-7: There is no reference to Champernowne number.

p11l-9: |alpha should be |\alpha|

Minor content remark: It bugs me a bit that this Kolmogorov complexity variant is "program"-dependent (program is here a finite state machine). Perhaps the complexity measure min{C_D(x)+sizeof(D) | D} is worth studying (not in this paper but for future work)?