

Normal numbers and algorithmic randomness: a historical sketch

V. Arvind*

The Institute of Mathematical Sciences, C.I.T. Campus, Chennai 600 113, India

What does it mean to say that a fixed infinite string is random? In this article we will attempt to trace the history of this question and the fundamental role of computability theory in our understanding of randomness. In particular, we will describe Turing's observations on the notion of normal numbers and their construction and how that connects up with algorithmic randomness.

Keywords: Algorithmic randomness, computability theory, infinite string, normal numbers.

Introduction

IN the early 20th century, when science was making great strides in physics and the foundations of modern mathematics were being deeply examined, a rigorous approach to randomness and probability was also being explored and developed. Historically, probability theory goes back at least to the 1600s to the work of Blaise Pascal in the study of games of chance. But probability theory became a field in its own right only in the 20th century.

One of the first serious attempts to formalize randomness was made by von Mises¹ in 1919. Intuitively, if an infinite sequence of 0's and 1's is random, then the number of 0's has to be the same as the number of 1's in the limit. However, this is clearly not a sufficient condition, because the sequence with 1's in odd positions and 0's in even positions has this property, but it is obviously not 'random' as it is easily 'predictable'. An infinite sequence, argued von Mises, should be considered random if the number of 0's and 1's is the same in the limit not only for the entire sequence but for every *admissible* infinite subsequence. Although von Mises's notion of admissibility was motivated by statistical tests, it lacked an adequate formal definition. It was Church² in 1940 who suggested that the right notion of admissible subsequences was precisely that of computable subsequences. Formal computability theory, which was introduced by Church and Turing³ in 1936, since then has been inevitable in any rigorous treatment of randomness. However, the definition of von Mises for random sequences turns out to be inadequate. Ville⁴, in 1936, showed there are

infinite binary sequences that are random in the sense of von Mises, but in every initial segment the number of zeros is more than the number of ones. Such infinite sequences form a measure zero set and therefore cannot be considered truly random.

Turning to another aspect of the question, we know that every real number in the unit interval $[0, 1]$ has an infinite binary expansion $0.a_1a_2\dots$. Thus real numbers can be identified with infinite sequences of 0's and 1's (or infinite sequences over some other base). However, we can *name* only *countably many* of the set of all real numbers. For, a name is just a finite sequence of a finite set of symbols (letters of the English alphabet, for example). Thus, all rational numbers have names (because all integers have names and rationals are ratios of integers). And some highly privileged irrational numbers like π and e have names. We can name some more numbers indirectly. For instance, we can talk of the positive root of $x^2 - 2$ or the real root of $x^3 - 5$. The problem is that the set of all names, being finite words over a finite alphabet, is *countable* and the set of all real numbers is *uncountable*. In terms of measure theory, the set of nameable real numbers, for any given finite alphabet used for naming, is a *measure zero* set. The rest of the real numbers, which is almost *all* of them, cannot be named. It is tempting to define random real numbers as those that cannot be named. However, we need to be careful about the notion of nameability itself, which is something ambiguous. For instance, consider all numbers that are nameable in English and order these names ordered lexicographically and in increasing order of their lengths. Then, 'the first real number that is not nameable in a thousand words' can be 'named' in just 12 words! This is known as Berry's paradox.

Let us consider Bernoulli trials: we repeatedly flip an unbiased coin and record the outcome 'heads' as 1 and 'tails' as 0. If we repeat this ad infinitum, we obtain an infinite binary sequence which corresponds to a real number in $[0, 1]$. Thus, we perform this *random experiment* of coin-flips and the *outcomes* are infinite binary sequences. But can we talk of a given binary sequence being random? An intuitive notion is that a random sequence is *unpredictable*, in the sense that there are no patterns in it. However, this needs to be carefully formalized. For instance, the infinite string of 0's is obviously not random. On the other hand, the infinite binary

*e-mail: arvind@imsc.res.in

expansion of an irrational number like $\sqrt{2}$ does not reveal any easily predictable pattern, although it is completely predictable!

Randomness and normal numbers

What then are random strings? In this section we will discuss this question further. Suppose we ask both Alice and Bob to produce a 100-bit random string each. Both Alice and Bob perform 100 independent unbiased coin flips to produce strings x and y respectively. Suppose x is a string of all zeros and y is a string with a fair distribution of 0's and 1's. Although both x and y are equally likely (they both occur with probability 2^{-100}), an observer is more likely to accept y as a random string than x ! To quote Laplace: 'In the game of heads and tails, if head comes up a hundred times in a row then it appears to us extraordinary ...', because, intuitively, those sequences in which we observe a rule that is easy to grasp are rare.

Thus, in random infinite binary sequences one would expect that every block of r bits appears with limiting frequency 2^{-r} . This was proposed by Emile Borel⁵ in 1909, earlier than von Mises, as the right notion of random sequences. This definition was motivated by the law of large numbers, and real numbers in $(0, 1)$ whose binary expansion are such sequences are called *normal numbers*.

Definition 1. *A real number in $[0, 1]$ is called a normal number in base b if its infinite expansion in base b is very balanced: every block of digits of length r occurs with limit frequency b^{-r} , for every positive integer r .*

An absolutely normal number, also called just normal number, is one that is normal in every base.

Borel proved that the set of nonnormal numbers has Lebesgue measure 0. Borel's notion of normality is clearly a desirable property for random sequences, but are all normal numbers random? Unfortunately, they need not be random. Champernowne⁶ first showed that the number 0.123...9101112... is normal in base 10, but it is clearly not random because there is an algorithm for generating the sequence making it completely predictable. Similarly, Copeland and Erdős⁷ showed that the expansion 0.235... obtained by listing down all primes in base 10 is normal in base 10. Although this rules out normality in specific bases as the right notion of randomness, it remains a challenging open question whether well-known natural examples of real numbers like π , e , $\sqrt{2}$, $\ln 2$, etc. are normal or not in some base.

The other major question was whether we can find numbers that are normal in *every base*. Or perhaps the notion of absolutely normal numbers is the right notion of randomness. Thus the question of finding *explicit* numbers that are normal in every base assumes importance.

What does explicit mean in this context? Numbers like π , e , $\sqrt{2}$ and $\ln 2$ are clearly explicit: we know them in the sense that we know their properties and know infinite series converging to them. In other words, these are *computable numbers*.

So, what are computable numbers, formally?

Definition 2. *Fix a programming language, say C , for the discussion. A real number $x \in (0, 1)$ is computable if there is a C program for x that, on input n , outputs the n th bit in the binary expansion of x .*

The definition of computable numbers is robust. A number x is computable in base 2 if it is computable in any other base b . Moreover, by the Church–Turing thesis, for any conceivable programming language we will obtain the same set of computable real numbers. This notion of computable numbers, of course, had to wait until the Turing's work formalizing computability.

Turing's construction of normal numbers

In particular, Turing interpreted 'explicit' as computable and made precise Borel's question as asking if there are computable normal numbers. Since computable reals form a countably infinite set (we can enumerate all C programs), they form a measure zero set. Hence we would not expect random numbers to be computable.

In an unpublished manuscript titled 'A note on normal numbers' (handwritten on the back of a copy of his 1936 'On computable numbers, with an application to the Entscheidungsproblem' paper), Turing outlined a construction of computable normal numbers. His was the first satisfactory construction of computable normal numbers. The manuscript was discovered much later in the 1960s. There were earlier attempted constructions of Lebesgue (in 1909) and Sierpinski (in 1917) that were inadequate. In this section we will outline the main ideas in Turing's work based on Becher *et al.*⁸

Turing's first observation is a combinatorial result. Fix the alphabet $\Sigma = \{0, 1, \dots, b-1\}$ for base b representation. Let $n_{b,r}(k, i)$ denote the number of strings in Σ^k in which a fixed string $w \in \Sigma^r$ occurs exactly i times. Clearly, this number is independent of the choice of w . For example

$$n_{b,1}(k, i) = \binom{k}{i} (b-1)^{k-i}.$$

As is well known, for a random string in Σ^k the expected number of occurrences of each letter in Σ is k/b and the actual number of occurrences is tightly concentrated around its expected value. Turing first generalizes this concentration bound.

Lemma 3. For a base $b \geq 2$ and $(6r/k) < \varepsilon \leq b^{-r}$,

$$\sum_{i: |i - kb^{-r}| \geq \varepsilon k} n_{b,r}(k, i) < 2rb^{k+2r-2} e^{-kb^r \varepsilon^2 / 6r}.$$

Let $S(x, b, w, k)$ denote the number of occurrences of a string w in the first k digits of the base b expansion of x . Let

$$A_k = \bigcap_{2 \leq b \leq 2^{\sqrt{6k/4}}} \bigcap_{1 \leq r \leq \sqrt{\ln k/4}} \bigcap_{w \in \Sigma^r} \{x \mid |S(x, b, w, k) - kb^{-r}| \leq k^{15/16}\}.$$

Then, since $\{x \mid |S(x, b, w, k) - kb^{-r}| \leq k^{15/16}\}$ is a finite union of intervals with rational end-points, A_k is also a finite union of intervals with rational endpoints. Let $\mu(A_k)$ denote its Lebesgue measure (i.e. the sum of the lengths of the said intervals). We can estimate it as follows.

Proposition 4. For all k larger than a constant k_0

$$\mu(A_k) \geq 1 - \frac{1}{k} + \frac{1}{k-1}.$$

Theorem 5 (Turing). For any $\ell \geq k_0$, $\bigcup_{k \geq \ell} \overline{A_k}$ contains all numbers that are not absolutely normal.

From A_k , Turing defines another collection of intervals $E_{k,n}$, where

$$E_{k,0} = (0, 1) \quad \text{and} \quad E_{k,n} = A_{k2^{2n+1}} \cap E_{k,n-1} \cap (\beta_n, 1),$$

where β_n is the unique rational so that $\mu(E_{k,n}) = 1 - (1/k) + (1/k2^{2n+1})$. Let $E_k = \bigcap_{n \geq 0} E_{k,n}$. Since $E_{k,n}$ is a monotonically decreasing sequence of sets with limit E_k , it follows that $\mu(E_k) = 1 - 1/k$. Furthermore, since $E_k \subseteq \bigcap_{\ell \geq k} A_\ell$, it follows that E_k contains only normal numbers.

A computable normal number

Turing's algorithm for computing a normal number $x = 0.x_1x_2x_3\dots$ works in stages. Choosing the first bit x_1 of x amounts to halving the unit interval $(0, 1)$. Likewise, every subsequent choice x_i , $i \geq 2$ will keep halving the interval and the construction needs to ensure that the intersection of the current interval with the set of normal numbers always has a positive measure. Suppose the interval at the n th stage is I_n when we have already fixed the first n bits x_1, x_2, \dots, x_n . Inductively, assume that

$$\mu(E_{k,n} \cap I_n) > \frac{1}{k2^{2n}}.$$

Now, notice that $E_{k,n+1} \cap I_n$ contains $(E_{k,n} \cap I_n) \setminus (E_{k,n} \setminus E_{k,n+1})$. Furthermore

$$\mu(E_{k,n} \setminus E_{k,n+1}) = \frac{1}{k2^{2n+1}} - \frac{1}{k2^{2(n+1)+1}}.$$

Putting it together, it follows that

$$\mu(E_{k,n+1} \cap I_n) > \frac{1}{k2^{2n+2}}.$$

Now, we partition I_n into two halves $I_n = I_n^0 \cup I_n^1$ by setting $x_{n+1} = 0$ and $x_{n+1} = 1$ respectively. Clearly for some $b \in \{0, 1\}$ we have

$$\mu(E_{k,n+1} \cap I_n^b) > \frac{1}{k2^{2n+2}}.$$

Letting $I_{n+1} = I_n^b$ we have established the induction step. Hence the construction yields a computable normal number x .

Algorithmic randomness

Turing's construction of the set $E_k = \bigcap_{n \geq 0} E_{k,n}$ which contains only normal numbers and $\mu(E_k) = 1 - 1/k$ can be interpreted as constructing the computable sequence $\overline{E_k}$ which has measure zero in the limit and contains all non-normal numbers. This is an example of a Martin-Löf test. Martin-Löf's idea⁹ is basically a formalization of the notion of statistical tests for randomness using computability theory. He essentially developed the rudiments of a computable version of measure theory by defining what it means for a set to be of *computable measure* one or zero. The basic open sets are intervals of the form $J_w \subset (0, 1)$ consisting of all numbers of the form $0.wx_1x_2\dots$ for arbitrary $x_i \in \{0, 1\}$ for words $w \in \{0, 1\}^*$. A set A is *computably null* if for each $\varepsilon > 0$ we can find a computable sequence of words w_k such that $A \subseteq \bigcup J_{w_k}$ and $\sum_{k > 0} 2^{-|w_k|} \leq \varepsilon$. Computable measure one sets are just the complements of null sets.

Definition 6. A Martin-Löf statistical test consists of a computable sequence of intervals $\{J_w(m,n)\}$ for words $w(m, n) \in \{0, 1\}^*$ such that for each m it holds that $\mu(\bigcup_n J_w(m, n)) \leq 2^{-m}$.

An alleged random sequence x fails the test at confidence level m if $x \in J_w(m, n)$ for some n . If for some m_0 the sequence x passes the test for all $m \geq m_0$, then x is said to pass the test.

A sequence is Martin-Löf random if it passes all Martin-Löf statistical tests.

Since there are countably many algorithms, there are countably many Martin-Löf statistical tests. Since the union of computable null sets is also a computable null set, as shown by Martin-Löf, it follows that there is a *universal* statistical test for randomness.

The set of Martin-Löf random sequences (equivalently, Martin-Löf random reals) forms a measure one subset of $(0, 1)$ and each of its elements is noncomputable. But how do we know that this is the right notion of randomness? Just as in the 1930s many different formal definitions of computable functions were proposed by Gödel, Church, Turing, Markov, Kleene, Post, etc. and all of these definitions coincided giving us the so-called Church–Turing thesis, a similar phenomenon occurred for the definition of algorithmic randomness which we now explain.

Three approaches to randomness

Statistical approach: This is the intuitive approach to defining randomness that we started out with with von Mises and Borel, culminating in Martin-Löf’s definition of randomness. The intuitive idea is that random sequences should not have recognizable patterns, i.e. have recognizable rare properties. Computable null sets are precisely these rare properties as formalized by Martin-Löf.

Incompressibility approach: The idea here is that recognizable patterns in a sequence means the sequence can be compressed. Hence, random sequences are the incompressible sequences. This approach was pioneered by Kolmogorov, and pursued by Levin and Chaitin in the 1960s.

Gambling approach: The roots of probability theory go back to analysing odds in gambling. A random sequence is one against which betting strategies cannot succeed more often than fail. Intuitively, there is always some betting strategy that can exploit rare patterns in a nonrandom sequence. For random sequences, no computable betting strategy can make infinite money by predicting successive bits. This is the martingale approach started by Ville in 1939 and developed by Schnorr 1971.

We now explain these two approaches in some more detail.

Incompressibility

We recall that in his seminal 1936 work Turing defined the (Turing) machine model. A fundamental concept was that of universal Turing machines. These are machines that can take the description of any Turing machine and simulate its computation on a given input, analogous to a compiler (or operating system). Turing constructed universal Turing machines in his paper.

For any finite string x its plain *Kolmogorov complexity* $C(x)$, w.r.t. a universal Turing machine U , is the length of the smallest program σ such that $U(\sigma) = x$. Thus, σ is a compressed form of x . For example, if x is the binary

string 0^{2^n} , then there is a program τ of length $\log_2 n + c$, where τ consists of the binary description of n along with a program of size c with instructions to produce the string 0^{2^n} from n . Note that c is a constant independent of n . Thus

$$C(0^{2^n}) \leq \log_2 n + c.$$

For any string x , clearly $C(x) \leq |x| + c$, because a program τ that contains the entire string x with instructions to print it out will suffice.

For a fixed universal Turing machine U , consider $C(x)$ for all x in the set $\{0, 1\}^n$. Clearly, the number of strings such that $C(x) \leq n - d$ for any given d is bounded by 2^{n-d+1} because each program of length at most $n - d$ can produce at most one string $x \in \{0, 1\}^n$. Thus, for each constant d , at least $2^n - 2^{n-d+1}$ strings x of length n exist such that

$$C(x) \geq |x| - d.$$

If $C(x) \geq |x| - d$ then x is said to be *d-incompressible*.

Although the C -measure is good for several purposes, it turns out that in order to define random sequences using Kolmogorov complexity we need a somewhat different measure. It is a refinement of $C(x)$ and is denoted by $K(x)$. The measure $K(x)$ is the Kolmogorov complexity of string x with respect to only those universal Turing machines U whose domain is prefix-free. Now we are ready to define a random sequence (or a real in $(0, 1)$) using incompressibility. A sequence $0.x_1x_2\dots$ is Kolmogorov random if there is some constant c such that for every prefix $x_1x_2\dots x_n$ of the sequence

$$K(x_1x_2\dots x_n) \geq n - c.$$

The following result shows that Martin-Löf random reals coincide with incompressible reals.

Theorem 7 (Schnorr 1971). *The following are equivalent for a real number $x = 0.x_1x_2x_3\dots$*

1. x is Martin-Löf random.
2. There is a positive integer c such that for all n

$$K(x_1x_2\dots x_n) \geq n - c.$$

Since the two approaches are different and yet the notions are equivalent, this increases our confidence that we have hit upon the correct notion of random sequences. We next discuss yet another approach.

The Gambler’s approach

Here the approach is based on betting strategies. The intuitive idea of a random sequence is one against which

no betting strategy can make an unbounded amount of money by predicting successive bits. Formally, a *martingale or betting strategy* is a function $d : \{0, 1\}^* \rightarrow \mathcal{R}^+$ such that $d(\varepsilon) > 0$ and

$$d(y) = \frac{d(y0) + d(y1)}{2},$$

for every finite string y . A martingale *succeeds* on a real number x if $\limsup d(x|n) = \infty$.

The definition can be interpreted as betting on the successive bits of the sequence $x = 0.x_1x_2\dots$. The gambler starts with some capital $d(\varepsilon) > 0$ before seeing any bit of x . At the i th step, after he has seen $x_1x_2 \dots x_{i-1}$ he bets a certain fraction $\alpha d(x_1x_2 \dots x_{i-1})$ on $x_i = 0$ and the remainder $(1 - \alpha)d(x_1x_2 \dots x_{i-1})$ on $x_i = 1$. The amount bet on the correct value will get doubled and the amount on the wrong value will be lost. Let $z = x_1x_2 \dots x_{i-1}$. Then, for the outcome 0 his capital will be $d(z0) = 2\alpha d(z)$ and for the outcome 1 his capital would be $d(z1) = 2(1 - \alpha)d(z)$. The actual strategy is the choice of $\alpha = (d(z0))/(2d(z))$.

The next theorem due to Ville shows that betting strategies capture the null sets (subsets of $(0, 1)$ of Lebesgue measure zero).

Theorem 8 (Ville 1939). *A set $A \subseteq (0, 1)$ has Lebesgue measure zero precisely if there is a martingale d that succeeds on all $x \in A$.*

It turns out, as shown by Schnorr¹⁰, that with a suitable definition of ‘computable’ betting strategies, such strategies succeed on precisely the computable null sets contained in $(0, 1)$. Hence we have the following equivalence theorem.

Theorem 9. *For any random sequence $x = x_1x_2 \dots \in (0, 1)$, the following statements are equivalent.*

1. *The real number $x \in (0, 1)$ is Martin-Löf random.*
2. *No computable martingale succeeds on the number x .*
3. *There is a positive integer c such that for all n*

$$K(x|n) \geq n - c.$$

Thus all three definitions of randomness coincide, tied together through computability. This is all very fine, but can we give an example of an infinite random sequence? Random sequences in $(0, 1)$ form a measure one set. Can we not point out even one of them? We conclude this section by defining Chaitin’s Ω (ref. 11). Fix a prefix-free universal Turing machine U and the alphabet, say $\{0, 1\}$. Let H denote the set of all programs $p \in \{0, 1\}^*$ such that the simulation by U of the program p on blank tape halts.

$$\Omega = \sum_{p \in H} 2^{-|p|}.$$

Since U is prefix-free, $\Omega \leq 1$. Since there are programs that halt and programs that do not halt, $0 < \Omega < 1$. It turns out that the expansion of Ω is a random sequence.

Normal numbers revisited

We discuss normal numbers again in this last section. As we have seen computable functions play a crucial role in the definition and understanding of random sequences. Since normal numbers were introduced by Borel in an attempt to define random sequences, it is satisfying to note that normal numbers are indeed random with respect to a weaker model of computation, namely finite automata. Finite automata are a very restricted version of Turing machines: they have a finite state control (like Turing machines) but are *memoryless*. Finite automata are well studied in computer science (e.g. see ref. 12). In other words, finite automata are to normal numbers exactly what computable functions are to random sequences.

The first connection between finite automata and normal numbers was noted by Agafonov¹³. Suppose $x = 0.x_1x_2 \dots$ is a real number in $(0, 1)$. Given any deterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F)$, we can define a subsequence x_{i_1}, x_{i_2}, \dots from the x_i s by running the automaton on the sequence x and outputting only those bits x_{i_j} when M enters an accepting state in F . This yields a new real number $y_M = 0.x_{i_1}x_{i_2} \dots$ defined by the automaton M from x .

It turns out, quite interestingly, that the number x is normal if and only if the number y_M is normal for every deterministic finite automaton M that outputs an infinite subsequence $x_{i_1}x_{i_2} \dots$.

It turns out that we can capture precisely normal numbers using betting strategies that are computable by finite automata. We can also capture normal numbers using an appropriate notion of incompressibility.

Definition 10. *A betting strategy computable by a finite automaton is defined by a DFA over alphabet, say Σ , where each state q has a fraction $0 \leq \alpha_a(q) \leq 1$ for each letter $a \in \Sigma$ so that $\sum_{a \in \Sigma} \alpha_a(q) = 1$, where $\alpha_a(q)$ is the fraction of the capital bet by the state q that the next letter is an a . This gives rise to the function $d : \{0, 1\}^* \rightarrow \mathcal{R}^+$ such that $d(\varepsilon) > 0$ and*

$$d(y) = \frac{d(y0) + d(y1)}{2},$$

for every finite string y . The strategy succeeds on a sequence $x = x_1x_2 \dots$ if $\limsup d(x_1x_2 \dots x_n) = \infty$.

The following was shown by Schnorr and Stimm¹⁴, and Bourke *et al.*¹⁵.

Theorem 11. *A number $x \in (0, 1)$, $x = 0.x_1x_2 \dots$ is normal if and only if there is no betting strategy computable by a finite automaton that succeeds on x .*

Just as prefix-free Kolmogorov complexity captures random sequences: random sequences are the Kolmogorov incompressible sequences, Ziv and Lempel¹⁶ showed that in the setting of finite automata, incompressibility coincides with normal numbers.

Definition 12. *A finite-state compressor is a deterministic finite automation with output strings (including ϵ) labeling each state transitions. A lossless compressor is a finite-state compressor whose input string can be recovered uniquely from the output string and final state.*

The finite-state compressor C is said to compress a number $x = 0.x_1x_2 \dots$ if

$$\liminf \frac{C(x_1x_2 \dots x_n)}{n} < 1.$$

We state the formal result of Ziv and Lempel¹⁶.

Theorem 13. *A real number $x \in (0, 1)$, where $x = 0.x_1x_2 \dots$ is normal if and only there is no finite-state compressor for x .*

1. von Mises, R., Grundlagen der Wahrscheinlichkeitsrechnung. *Math. Z.*, 1919, **5**, 52–89.

2. Church, A., On the concept of a random sequence. *Bull. AMS*, 1940, **46**, 130–135.
3. Turing, A. M., On computable numbers with an application to the Entscheidungsproblem. *Proc. London Math. Soc. Ser. 2*, 1936, **42**, 230–265.
4. Ville, J., Etude Critique de la notion de collectif. In *Gauthier-Villars*, Paris, 1939.
5. Borel, E., Les probabilités dénombrables et leurs applications arithmétiques. *Rend. Circ. Mat. Palermo*, 1909, **27**, 247–271.
6. Champernowne, D. G., The construction of decimals normal in the scale of ten. *J. London Math. Soc.*, 1933, **8**(4), 254–260.
7. Copeland, A. H. and Erdős, P., Note on normal numbers. *Bull. AMS*, 1946, **52**(10), 857–860.
8. Becher, V., Figueira, S. and Picchi, R., Turing's unpublished algorithm for normal numbers. *Theor. Comput. Sci.*, 2007, **377**, 126–138.
9. Martin-Löf, P., The definition of random sequences. *Inf. Control*, 1966, **9**, 60–619.
10. Schnorr, C. P., A unified approach to the definition of random sequences. *Math. Syst. Theory*, 1971, **5**, 246–258.
11. Chaitin, G., A theory of program size formally identical to information theory. *J. ACM*, 1975, **22**, 329–340.
12. Hopcroft, J. E. and Ullman, J. D., *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.
13. Agafonov, V. N., Normal sequences and finite automata. *Sov. Math. Dokl.*, 1968, **9**, 324–325.
14. Schnorr, C. P. and Stimm, H., Endliche automaten und zufallsfolgen. *Acta Inf. I*, 1972, 345–359.
15. Bourke, C., Hitchcock, J. and Vinodchandran, N. V., Entropy rates and finite state dimension. *Theor. Comput. Sci.*, 2005, **349**(3), 392–406.
16. Ziv, J. and Lempel, A., Compression of individual sequences via variable-rate coding. *IEEE Trans. Inf. Theory*, 1978, **24**(5), 530–536.
17. Turing, A. M., A note on normal numbers. In *Collected Works of A. M. Turing: Pure Mathematics* (ed. Britton), North Holland, 1992, pp. 263–265.