Randomness and Computation

Rod Downey
School of Mathematics and Statistics, Victoria University, PO Box 600, Wellington, New Zealand
rod.downey@vuw.ac.nz

Abstract

This article examines work seeking to understand randomness using computational tools. The focus here will be how these studies interact with classical mathematics, and progress in the recent decade. A few representative and easier proofs are given, but mainly we will refer to the literature. The article could be seen as a companion to, as well as focusing on developments since, the paper "Calibrating Randomness" from 2006 which focused more on how randomness calibrations correlated to computational ones.

1 Introduction

The great Russian mathematician Andrey Kolmogorov appears several times in this paper, First, around 1930, Kolmogorov and others founded the theory of probability, basing it on measure theory. Kolmogorov's foundation does not seek to give any meaning to the notion of an individual object, such as a single real number or binary string, being random, but rather studies the expected values of random variables. As we learn at school, all strings of length n have the same probability of 2^{-n} for a fair coin. A set consisting of a single real has probability zero. Thus there is no meaning we can ascribe to randomness of a single object.

Yet we have a persistent intuition that certain strings of coin tosses are less random than others. The goal of the theory of algorithmic randomness is to give meaning to randomness content for individual objects. Quite aside from the intrinsic mathematical interest, the *utility* of this theory is that using such objects instead distributions might be significantly simpler and perhaps giving alternative insight into what randomness might mean in mathematics, and perhaps in nature.

College Publications, London 2020

2 Historical roots

2.1 Borel and von Mises

Predating the work of Kolmogorov are early attempts to answer this kind of question by providing notions of randomness for individual objects. The modern theory of *algorithmic randomness* realizes this goal. One way to develop this theory is based on the idea that an object is random if it passes all relevant "randomness tests". That is, for a desired level of randomness, we would have computational tests for which we wold regard a real or string[1] as random it is "passed" such tests. The idea is that we could not distinguish the real passing the test from one that was "really" random.

For example, by the law of large numbers, for a random real X, we would expect the number of 1's in the binary expansion of X to have limiting frequency $\frac{1}{2}$. That is, we would expect to have

$$\lim_{n\to\infty} \frac{|\{j < n : X(j) = 1\}|}{n} = \frac{1}{2}.$$

Moreover, we would expect X to be *normal* to base 2, meaning that for any binary string σ of length k, the occurrences of σ in the binary expansion of X should have limiting frequency 2^{-k}. Since base representation should not affect randomness, we would expect X to be normal in this sense no matter what base it were written in, so that in base b the limiting frequency would be b^{-k} for a string σ of length k. Thus X should be what is known as *absolutely normal.*

The concept of (absolute) normality is due to Borel [26] in around 1909. It remains a thriving area of number theory, which has had significant advances; particularly via interactions with algorithmic randomness as we later see. In an attempt to characterize randomness, normality was extended by von Mises [131] in 1919. Von Mises suggested the following definition of randomness for individual binary sequences. A *selection function* is an increasing function $f : \mathbb{N} \to \mathbb{N}$. We think of $f(i)$ as the ith place selected in forming a subsequence of a given sequence. (For the definition of normality above, where we consider the entire sequence, $f(i) = i$.) Von Mises suggested that a sequence $a_0a_1\dots$ should be random if any selected subsequence $a_{f(0)}a_{f(1)}\dots$ is normal.

The reader will immediately notice the following problem: sequence X with infinitely many 1's, *post hoc* we could let f select the positions where 1's occur, and X would fail the test determined by f. However, it does not seem reasonable

[1] For this paper we consider "reals" as members of 2^ω and strings in $2^{<\omega}$ unless otherwise specified. We will denote the j-th bit of α as $\alpha(j)$, and first n bits of a real (or string) α will be denoted by $\alpha \restriction n$.

to be able to choose the testing places *after* selecting an X. The question is then: What kinds of selection functions should be allowed, to capture the intuition that we ought not to be able to sample from a random sequence and get the wrong frequencies? The statistician Wald [132, 133] (also famous for his analysis of bullet damage on warplanes in World War 2) showed that for any *countable* collection of selection functions, we could construct a real passing the tests they generate.

It is reasonable to regard prediction as a computational process, and hence restrict ourselves to *computable* selection function[2] The reader will note that Borel's and von Mises' work predates the events the early 1930's where the notion of a computable function was clarified by Church, Kleene, Post and famously Turing [128]. We remark that it is clear that Borel had a very good intuitive understanding of what a computable process was; see Avigad and Brattka for discussion of the development of computable analysis. But there was no formal clarification until the Church-Turing Thesis.

Indeed, this suggestion to use computable selection functions was eventually made by Church [38] in 1940, and this notion is now known as *Church Stochasticity.* As we will see, von Mises' approach had a more significant flaw, but we can build on its fundamental idea: Imagine that we are judges deciding whether a sequence X should count as random. If X passes all tests we can (in principle) devise given our computational power, then we should regard X as random since, as far as we are concerned, X has all the expected properties of a random object. We will use this intuition and the apparatus of computability and complexity theory to describe notions of *algorithmic* randomness.

Aside from the intrinsic interest of such an approach, it leads to useful mathematical tools. Many processes in mathematics are computable. Indeed any process from "real life" would surely be computable. Hence the *expected behavior* of such a process should align itself with the behavior obtained by providing it with an *algorithmically random input.* Hence, instead of having to analyze the relevant distribution and its statistics, we can simply argue about the behavior of the process on a single input. For instance, the expected number of steps of a sorting algorithm should be the same as that for a single algorithmically random input. We could also be more fine-grained and seek to understand exactly "how much" randomness is needed for certain typical behaviors to arise. (See Section 5.)

As we will discuss, algorithmic randomness also goes hand in hand with other parts of algorithmic information theory, such as Kolmogorov complexity, and has ties with notions such as Shannon entropy and fractal dimension.

[2] Indeed for practical applications, we might restrict ourselves to *polynomial time* or even *automatic* selections.

2.2 Some basic computability theory

Given that this is an expository paper in a Logic volume, we will assume that the reader is more or less cognoscent with the rudiments of classical computability theory. Thus we will give a brief reprise of the concepts we will be using Given that this is an expository paper in a Logic volume, we will assume that the reader is more or less cognoscent with the rudiments of classical computability theory. Thus we will give a brief reprise of the concepts we will be using In the 1930's, Church, Gödel, Kleene, Post, and most famously Turing [128] gave equivalent mathematical definitions capturing the intuitive notion of a computable function, leading to the *Church-Turing Thesis*, which can be taken as asserting that a function (from $\mathbb{N}$ to $\mathbb{N}$, say) is computable if and only if it can be computed by a Turing machine[3]. It has also become clear that algorithms can be treated as data, and hence that there is a *universal Turing machine*, i.e., there are a listing $\Phi_0, \Phi_1, \ldots$ of all Turing machines and a single algorithm that, on input $\langle e, n\rangle$ computes the result $\Phi_e(n)$ of running Φ_e on input n.[4]

It is important to note that a Turing machine might not halt on a given input, and hence the functions computed by Turing machines are in general *partial.* Indeed, as Turing showed, the *halting problem* "Does the eth Turing machine halt on input n?" is algorithmically unsolvable. Church and Turing famously showed that Hilbert's *Entscheidungsproblem* (the decision problem for first-order logic) is unsolvable, in Turing's case by showing that the halting problem can be coded into first-order logic. Many other problems have since been shown to be algorithmically unsolvable by similar means.

We write $\Phi_e(n)\downarrow$ to mean that the machine Φ_e eventually halts on input n. Then $\emptyset' = \{\langle e, n\rangle : \Phi_e(n)\downarrow\}$ is a set representing the halting problem. This set is an example of a noncomputable *computably enumerable* (*c.e.*) set, which means that the set can be listed (not necessarily in numerical order) by some algorithm.

[3] This definition can easily be transferred to other objects of countable mathematics. For instance, we think of infinite binary sequences as functions $\mathbb{N} \to \{0, 1\}$, and identify sets of natural numbers with their characteristic functions.

[4] The realization that such universal machines are possible helped lead to the development of modern computers. Previously, machines had been purpose-built for given tasks. In a 1947 lecture on his design for the Automated Computing Engine, Turing said, "The special machine may be called the universal machine; it works in the following quite simple manner. When we have decided what machine we wish to imitate we punch a description of it on the tape of the universal machine ... The universal machine has only to keep looking at this description in order to find out what it should do at each stage. Thus the complexity of the machine to be imitated is concentrated in the tape and does not appear in the universal machine proper in any way. ... [D]igital computing machines such as the ACE ... are in fact practical versions of the universal machine." From our contemporary point of view, it may be difficult to imagine how novel this idea was.

Another important notion is that of *Turing reducibility* (which we define for sets of natural numbers but is similarly defined for functions), where A is Turing reducible to B, written as $A \leqslant_{\mathrm{T}} B$, if there is an algorithm for computing A when given oracle access to B. That is, the algorithm is allowed access to answers to questions of the form "Is n in B?" during its execution. This notion can be formalized using Turing machines with oracle tapes, or by adding the characteristic function of B to the Kleene partial recursive functions. If $A \leqslant_{\mathrm{T}} B$, then we regard A as no more complicated than B from a computability-theoretic perspective. We also say that A is *B-computable* or *computable relative to B*. The pre-ordering $\leq_T$ naturally leads to an equivalence relation, where A and B are *Turing equivalent* if $A \leqslant_{\mathrm{T}} B$ and $B \leqslant_{\mathrm{T}} A$. The *(Turing) degree* of A is its equivalence class under this notion. As we know, if we examine exactly how the access mechanism works we get other reducibilities refining $\leq_T$. For example, $A \leq_m B$ means that there is a computable f such that $x \in A$ iff $f(x) \in B$. The polynomial miniaturization of this is central in computational complexity theory, as per Garey and Johnson [62]. Another reducibility of relevance to algorithmic randomness is *truth table* reducibility, where $A \leq_{tt} B$ means that $A \leq_T B$ via some procedure $\Phi^B = A$ such that for all oracles X, Φ^X is total.

In general, the process of allowing access to an oracle in our algorithms is known as *relativization*. As in the unrelativized case, we can list the Turing machines $\Phi_0^B, \Phi_1^B, \ldots$ with oracle B, and let $B' = \{\langle e, n\rangle : \Phi_e^B(n)\downarrow\}$ be the relativization of the halting problem to B. This set is called the *(Turing) jump* of B. The jump operation taking B to B' is very important in computability theory, one reason being that B' is the most complicated set that is still c.e. relative to B, i.e., B' is c.e. relative to B and every set that is c.e. relative to B is B'-computable. There are several other important classes of sets that can be defined in terms of the jump. For instance, A is *low* if $A' \leqslant_{\mathrm{T}} \emptyset'$ and *high* if $\emptyset'' \leqslant_{\mathrm{T}} A'$ (where $\emptyset'' = (\emptyset')'$). Low sets are in certain ways "close to computable", while high ones partake of some of the power of $\emptyset'$ as an oracle. These properties are invariant under Turing equivalence, and hence are also properties of Turing degrees. These concepts can be iterated, for example, $A^{(2)} = (A')'$ and the hierarchy of Turing degrees $\mathbf{0}', \mathbf{0}'', \ldots$ is called the *arithmetical hierarchy*, and transfinite iterations are called the *hyperarithmetical* hierarchy. See e.g. Downey and Hirschfeldt [47], Rogers [118], or Soare [124].

2.3 Martin-Löf randomness

Von Mises approach refined by Church to consider selection functions restricted to the computable ones. However, in 1939, Ville [130] showed that von Mises' approach cannot work in its original form, no matter what *countable* collection of selection functions we choose.

Theorem 2.1 (Ville [130]). *For any countable collection of selection functions, there is a sequence X that passes all von Mises tests associated with these functions, such that for every n, there are more 0's than 1's in $X \upharpoonright n$.*

I think if you went to a casino and were told there would *always* be more heads that tails you would not think the coin to be fair. We could try to repair von Mises' definition by adding further tests, reflecting statistical laws beyond the law of large numbers. But which ones? Ville suggested ones reflecting the law of iterated logarithms, which would take care of his specific example. But how could we know that further examples along these lines—i.e., sequences satisfying both von Mises' and Ville's tests, yet failing to have some other property we expect of random sequences—would not arise? For more of this, and a modern proof of Ville's Theorem see Downey and Hirschfeldt [47], where it is also shown that the law of iterated logarithms can be defeated.

The situation was finally clarified in the 1960's by Martin-Löf [96]. In probability theory, "typicality" is quantified using measure theory, leading to the intuition that random objects should avoid null sets. Martin-Löf noticed that tests like von Mises' and Ville's can be thought of as *effectively* null sets. Instead of considering specific tests based on particular statistical laws, we should consider *all* possible tests corresponding to some precisely defined notion of effectively null set. The restriction to such a notion gets around the problem that no sequence can avoid being in *every* null set. We will see later that this idea was anticipated by Turing around 1939 in unpublished notes for work on normality.

To give Martin-Löf's definition, we work for convenience in Cantor space 2^ω, whose elements are infinite binary sequences. The choice of base is not important. For example, all of the notions of randomness we consider are enough to ensure absolute normality. The basic open sets of Cantor space are the ones of the form $[\sigma] = \{X \in 2^\omega : X \text{ extends } \sigma\}$ for $\sigma \in 2^{<\omega}$, where $2^{<\omega}$ is the set of finite binary strings. The uniform measure λ on this space is obtained by defining $\lambda([\sigma]) = 2^{-|\sigma|}$. We say that a sequence $T_0, T_1, \ldots$ of open sets in 2^ω is *uniformly c.e.* if there is a c.e. set $G \subseteq \mathbb{N} \times 2^{<\omega}$ such that $T_n = \bigcup\{[\sigma] : (n, \sigma) \in G\}$.

Definition 2.2. A *Martin-Löf test* is a sequence $T_0, T_1, \ldots$ of uniformly c.e. open sets such that $\lambda(T_n) \leqslant 2^{-n}$. A sequence X *passes* this test if $X \notin \bigcap_n T_n$. A sequence is *Martin-Löf random* (*ML-random*) if it passes all Martin-Löf tests.

The intersection of a Martin-Löf test is our notion of effectively null set. Since there are only countably many Martin-Löf tests, and each determines a null set in the classical sense, the collection of ML-random sequences has measure 1. It can be shown that Martin-Löf tests include all the ones proposed by von Mises and Ville, in Church's computability-theoretic versions. Indeed they include all tests that are "computably performable", which avoids the problem of having to adaptively introduce more tests as more Ville-like sequences are found.

Martin-Löf's effectivization of measure theory allowed him to consider the laws a random sequence should obey from an abstract point of view, leading to a mathematically robust definition. As Jack Lutz said in a talk at the *7th Conference on Computability, Complexity, and Randomness* (during the Alan Turing Year programme in Cambridge, 2012),

> "Placing computability constraints on a nonconstructive theory like Lebesgue measure seems *a priori* to weaken the theory, but it may strengthen the theory for some purposes. This vision is crucial for present-day investigations of individual random sequences, dimensions of individual sequences, measure and category in complexity classes, etc."

2.4 The three approaches

ML-randomness can be thought of as the *statistician's approach* to defining algorithmic randomness, based on the intuition that random sequences should avoid having statistically rare properties. There are two other major approaches:

- The *gambler's approach*: random sequences should be unpredictable.
- The *coder's approach*: random sequences should not have regularities that allow us to compress the information they contain.

2.4.1 The Gambler's Approach

The gambler's approach may be the most immediately intuitive one. It was formalized in the computability-theoretic setting by Schnorr [120], using the idea that we should not be able to make arbitrarily much money when betting on the bits of a random sequence. The following notion is a simple special case of the notion of martingale from probability theory. (See [47, Section 6.3.4] for further discussion of the relationship between these concepts.)

Definition 2.3. A *martingale* is a function $f : 2^{<\omega} \to \mathbb{R}^{\geqslant 0}$ such that

$$f(\sigma) = \frac{f(\sigma 0) + f(\sigma 1)}{2}$$

for all σ. We say that f *succeeds* on X if $\limsup_{n\to\infty} f(X \upharpoonright n) = \infty$. We call this a *supermartingale* if for all σ,

$$f(\sigma) \geq \frac{f(\sigma 0) + f(\sigma 1)}{2}$$

We think of f as the capital we have when betting on the bits of a binary sequence according to a particular betting strategy. The displayed equation ensures that the betting is fair. Success then means that we can make arbitrarily much money when betting on X, which should not happen if X is random. By considering martingales with varying levels of effectivity, we get various notions of algorithmic randomness, including ML-randomness itself, as it turns out.

For example, if we insist that f is computable, we get a notion called *computable randomness.* Defining randomness for infinite sequences, by in complexity classes we might restrict f to be polynomial time, giving *polynomial time randomness* (Lutz [91]). We call a function f *left-c.e.* if there is a computable function $g(\cdot,\cdot)$ such that

- $\lim_s g(x,s)$ exists for all x, and
- $\lim_s g(x,s) = f(x)$, and
- $g(x, s+1) \geq g(x,s)$ for all x, s.

Right-c.e. reals are defined similarly, but using approximations from above.

Theorem 2.4 (Schnorr [120]). *X is ML-random iff no left c.e. (super-)martingale succeeds on it.*

The reader might wonder why we have "left c.e." in the characterization. The way to think about this is the following. We want to bet against sequences that we think are not random. Now as time evolves we will discover more and more facts. For example, is X was $\pi - e$, and we were given the binary expansion of this thoroughly computable number, which is therefor far from random, we would probably have trouble discerning the pattern. But as time went on as we computes more and more (partial) computable functions, we might discover a predictor yielding bits of X and hence we would like to place more capital on the bits of X.

Interestingly, this leads us to some basic problems which stubbornly remain open. First we might ask what happens if we have a partial computable betting function, but instead of betting on the first bit, then the second, etc we could bet on some bit, then, depending on the outcome, we could then bit somewhere we have not yet bet upon. This notion is called *non-monotonic* randomness.

Question 2.5 (Muchnik, Semenov and Uspensky [107])**.** Do ML-randomness and non-monotonic randomness coincide

The essence of the question is whether partial computable, rather than left c.e. methods suffice to define ML-randomness. In some sense they do if we replace martingales by "martingales processes" (a generalization of the martingale idea) as in Merkle, Mihailovíc, and Slaman [98]. Question 2.5 has been open since 1998. Another apparently difficult problem asks if *bias* is possible, which one would suspect that this is not possible. From [47], a *Kastergale* is a supermartingale together with a partial computable function $h : 2^{<\omega} \to \{0, 1\}$. Then if $h(\sigma) \downarrow [s]$ we will promise that for all $s' > s$, $g(\sigma h(\sigma), s') > g(\sigma, 1 - h(\sigma), s')$. That is we will always henceforth bias towards, e.g., 0 is $h(\sigma) = 0$.

Question 2.6. Is Kastergale-random (i.e. random for left-c.e. kastergales) the same as ML-random?

For some recent progress on these questions, we refer the reader to Barmpalias, Fang and Lewis-Pye [9].

It is not too difficult to show that ML-randomness is strictly stronger than computable randomness, but the difference is slight as quantified by computability theory:

Theorem 2.7 (Nies, Stephan, and Terwijn [112])**.** *If X is a computably random real and $X' \not\geq_T \emptyset''$ (X is not high), then X is ML-random. Every high degree contains a computably random real which is not ML-random.*

We will strengthen this result soon. Clearly there are many possible variations on the notion of martingale. For example, we might ask that there is a minimum possible bet (real casinos don't allow ε as a bet), or indeed we can only bet in some set of bets like $\{n_1, \ldots, n_k\}$ or $\mathbb{N}$. This leads to the interesting notion of integer-valued betting strategies, and we refer to Bienvenu, Stephan and Teutsch [24] for more on this. Later we will see that normality can be characterized by *automatic* martingales.

There are many other interesting levels of algorithmic randomness. Schnorr also introduced another notion related to the ML-randomness definition. He defined a notion now called *Schnorr randomness*, which is like the notion of computable randomness mentioned below Definition 2.3 but with an extra effectiveness condition on the rate of success of martingales. He also showed that X is Schnorr random iff it passes all Martin-Löf tests $T_0, T_1, \ldots$ such that the measures $\lambda(T_n)$ are uniformly computable (i.e., the function $n \mapsto \lambda(T_n)$ is computable in the sense of Section 5.4 below). We remark that it is also possible to give test characterizations of computable randomness but they are somewhat counter-intuitive. (See

[47].) It follows immediately from their definitions in terms of martingales that ML-randomness implies computable randomness, which in turn implies Schnorr randomness. It is more difficult to prove that none of these implications can be reversed.

Additionally to Theorem 2.7, Nies, Stephan and Terwijn [112] showed that Schnorr and computable randomness can be separated in the high degrees, but again coincide in the non-high ones.

2.4.2 The Coder's Approach

The coder's approach builds on the idea that a random string should have no short descriptions. For example, in describing 010101... (1000 times) by the brief description "print 01 1000 times", we are using regularities in this string to compress it. For a more complicated string, say the first 2000 bits of the binary expansion of e^{π}, the regularities may be harder to perceive, but are still there and can still lead to compression. A random string should have no such exploitable regularities (i.e., regularities that are not present in most strings), so the shortest way to describe it should be basically to write it out in full.

Again we see that Kolmogorov enters the picture: We formalize this using the well-known concept of Kolmogorov complexity. We can think of a Turing machine M with inputs and outputs in $2^{<\omega}$ as a description system. If $M(\tau) = \sigma$ then τ is an M-description of σ.

Definition 2.8. The *Kolmogorov complexity* $C_M(\sigma)$ of σ relative to M is the *length* of the shortest τ such that $M(\tau) = \sigma$.

Since we can enumerate all machines $\{M_e \mid e \in \mathbb{N}\}$, we can then take a universal Turing machine U, which emulates any given Turing machine with at most a constant increase in the size of programs. To wit, we would consider a machine U which on input $1^e0\sigma$ would run $M_e(\sigma)$. This is called universality by *adjugation.* We can then define the *(plain) Kolmogorov complexity* of σ as $C(\sigma) = C_U(\sigma)$. The value of $C(\sigma)$ depends on U, but only up to an additive constant independent of σ. We think of a string as random if its Kolmogorov complexity is close to its length.

For an infinite sequence X, a natural guess would be that X should be considered random if every initial segment of X is incompressible in this sense, i.e., if $C(X \upharpoonright n) \geqslant n - O(1)$[5]. However, plain Kolmogorov complexity is not quite the right notion here, because the information in a description τ consists not only of

[5] If, for example, $C(\sigma) \geq n \pm O(1)$, henceforth we will (mostly) use the economical notation $C(\sigma) \geq^{+} n$; and similarly for K below.

the bits of τ, but also its length, which can provide another $\log_2 |\tau|$ many bits of information. Indeed, Martin-Löf (see [90]) showed that it is not possible to have $C(X \upharpoonright n) \geqslant n - O(1)$: Given a long string ρ, we can write $\rho = \sigma\tau\nu$, where $|\tau|$ is the position of σ in the length-lexicographic ordering of $2^{<\omega}$. Consider the Turing machine M that, on input η, determines the $|\eta|$th string ξ in the length-lexicographic ordering of $2^{<\omega}$ and outputs $\xi\eta$. Then $N(\tau) = \sigma\tau$. For any sequence X and any k, this process allows us to compress some initial segment of X by more than k many bits.

There are several ways to get around this problem by modifying the definition of Kolmogorov complexity. The best-known one is to use prefix-free codes, which act like telephone numbers. That is, we restrict ourselves to machines M such that if $M(\tau)$ is defined (i.e., if the machine eventually halts on input τ) and μ is a proper extension of τ, then $M(\mu)$ is not defined. There are universal prefix-free machines, using the same method above, since we can enumerate the partial prefix-free machines. Then we can take such a machine U and define the *prefix-free Kolmogorov complexity* of σ as $K(\sigma) = C_U(\sigma)$. The roots of this notion be found in the work of Levin, Chaitin, and Schnorr, and in a certain sense—like the notion of Kolmogorov complexity more generally—even earlier in that of Solomonoff (see [47, 90]). As shown by Schnorr (see Chaitin [34]), it is indeed the case that the following theorem holds:

Theorem 2.9 (Schnorr). *X is Martin-Löf random if and only if $K(X \upharpoonright n) \geqslant^{+} n$.*

We remark that this shows that we our definitions are reasonably robust, in that all approaches yield the same ML-random reals. We remark that it is possible to give machine characterizations of computable and Schnorr randomness. For instance, we can call a machine M a *computable domain* machine if $\lambda\mathrm{dom}(M)$ is a computable real.

Theorem 2.10 (Downey and Griffiths [46]). *X is Schnorr random iff for all computable domain machines $K_M(X \upharpoonright n) \geq^{+} n$ for all n.*

We remark that there are other methods to capture ML-randomness using incompressibility. One is to use things akin to prefix-free complexity, like *process complexity* which asks that the action be *continuous.* That is we have machines M such that if $M(\sigma)\downarrow$ and $M(\tau)\downarrow$, and $\sigma \prec \tau$, then $M(\sigma) \preceq M(\tau)$. Using this Schnorr and earlier Levin (with an analogous concept) showed that again X is random iff its segments cannot be compressed. (see [47] for a discussion about this and similar compressors.) Day [40] gave a nice machine characterization of computable randomness using a kind of process machine as described below.

There are other varieties of Kolmogorov complexity, but C and K are the main ones. For applications[6], it often does not matter which variety is used. The following surprising result establishes a fairly precise relationship between C and K. Let $C^{(1)}(\sigma) = C(\sigma)$ and $C^{(n+1)}(\sigma) = C(C^{(n)}(\sigma))$.

Theorem 2.11 (Solovay [125])**.** $K(\sigma) = C(\sigma) + C^{(2)}(\sigma) \pm O(C^{(3)}(\sigma))$*, and this result is tight in that we cannot extend it to* $C^{(4)}(\sigma)$*.*

There is a simplified version of Solovay's original proof in [47] using a suggestion of Miller and a useful result known as Symmetry of Information.

Theorem 2.12 (Symmetry of Information for K[7]-Levin and Gács [60], Chaitin [34])**.** $K(\sigma, \tau) =^{+} K(\sigma) + K(\tau \mid \sigma, K(\sigma)) =^{+} K(\tau) + K(\sigma|\tau, K(\tau))$.

Using this and other techniques, Bauwens [13] gave some simpler proofs for Theorem 2.11.

There is a vast body of research on Kolmogorov complexity and its applications. We will discuss some of these applications below; much more on the topic can be found in the books of Li and Vitányi [90] (especially for applications) and Shen, Uspenskyi and Vereshchagin [122].

One notion of compression not to be found in [47], and largely forgotten, is the following.

Definition 2.13 (Kobayashi [85])**.**

1. Given $f : \mathbb{N} \to \mathbb{N}$, we say that X is *f-compressible* if there exists Y which computes X via an oracle Turing machine which queries, for each n, at most the first $f(n)$ digits of Y (i.e. the Y-use) for the computation of $X \upharpoonright n$.

2. We say that a real X is *Kobayashi incompressible* if it is not f-compressible for any function f such that $n - f(n)$ is unbounded.

A recent result shows that Kobayashi incompressibility actually coincides with ML-randomness.

Theorem 2.14 (Kobayashi incompressibility and Turing reductions-Barmpalias and Downey [12], Bienvenu)**.** *The following are equivalent:*

1. *X is Martin-Löf random;*

2. *For every Y with $X \leq_T Y$ the Y-use in any such computation of $X \upharpoonright n$ is bounded below by $n - c$ for some constant c and all n.*

[6] Particularly those involving effective fractal dimension we see later.

[7] There is also one for C.

Later we will look at other calibrations of randomness. For some there are characterizations along the Kobayashi-lines of the above such as for Kurtz randomness using stronger reducibilities than $\leq_T$.

3 Goals

There are several ways to explore the ideas introduced above. First, there are natural internal questions

- How do the various levels of algorithmic randomness interrelate?
- How do calibrations of randomness relate to the hierarchies of computability and complexity theory, and to relative computability?
- How should we calibrate partial randomness?
- Can a source of partial (algorithmic) randomness be amplified into a source that is fully random, or at least more random?

The books Downey and Hirschfeldt [47] and Nies [109] cover material along these lines up to about 2010.

We can also consider applications. Mathematics has many theorems that involve "almost everywhere" behavior. Natural examples come from ergodic theory, analysis, geometric measure theory, and even combinatorics. Behavior that occurs almost everywhere should occur at sufficiently random points. Using notions from algorithmic randomness, we can explore exactly *how much* randomness is needed in a given case. For example, the set of reals at which an increasing function is differentiable is null. How complicated is this null set, and hence, what level of algorithmic randomness is necessary for a real to avoid it (assuming the function is itself computable in some sense)? Is Martin-Löf randomness the right notion here? More specifically, suppose that I want to use randomness as a tool in some combinatorial algorithm. There are many such algorithms which ask for random seeds; for instance polynomial identity testing. What algorithmic level of source randomness is needed for applications to obtain results which are close to exact solutions? Also how does this theory relate to the well-developed theory of e.g. random graphs?

One recent example comes from an answer to of a question of Bollobas and of Kahane going back to 1965. In the introduction to his book [25] on random graphs, Bollobas motivates the use of probabilistic ideas in graph theory. He mentioned that earlier probabilistic application had been found in analysis via three seminal papers of Paley and Zigmund [113, 114, 115].

> "Paley and Zigmund (1930a,b,1932) had investigated random series of functions.One of their results was that if the real numbers c_n satisfy $\sum_{n=0}^{\infty} c_n^2 = \infty$ then $\sum_{n=0}^{\infty} \pm c_n \cos nx$ fails to be a Fourier-Lebesgue series for almost all choices of the signs. To exhibit a sequence of signs with this property is surprisingly difficult: indeed there is no algorithm known which constructs an appropriate sequence of signs from any sequence c_n with $\sum_{n=0}^{\infty} c_n^2 = \infty$."

We remark that an indentical question can be found even earlier in the 1968 version of Kahane's book (most recently, [80], page 47), on random trigonometric series:

> "A surprising fact is that nobody knows how to construct these signs explicitly, but a random choice works."

In recent work, Downey, Greenberg and Tangarra [45] showed that this question has a positve answer by showing that the collection of signs where the series *converges* forms a Kurtz null test (i.e the complement of a c.e. open set of measure 1)[8]. Hence by general theorems about Kurtz null tests we know that there is a computable real which succeeds on this test. There is a huge amount of largely unexplored work on random trigonometric series, some of which is explored in [45], and earlier [116].

We can also use the idea of assigning levels of randomness to individual objects to prove new theorems or give simpler proofs of known ones. We give some examples later, especially in the area of Hausdorff dimension theory.

3.0.1 The Incompressibility Method

Early examples of this method tended to use Kolmogorov complexity and what is called the "incompressibility method". For instance, in 1975, Chaitin [33] (see also [86]) famously used Kolmogorov complexity to give a proof of a version of Gödel's First Incompleteness Theorem, by showing the following:

Theorem 3.1 (Chaitin [33]; also Barzdins)**.** *For any sufficiently strong, computably axiomatizable, consistent theory T, there is a number L such that T cannot prove that $C(\sigma) > L$ for any given string σ*[9]*.*

Proof. (Sketch-Kritchman and Raz [86]) (For this proof, C or K are equally usable.) Let L be a large enough integer. Assume for a contradiction that, for some

[8] We look at Kurtz randomness, a notion of randomness weaker than ML-randomness, later in the present article.

[9] This also follows by interpreting an earlier result of Barzdins; see [90, Section 2.7]).

integer x, there is a proof for the statement "$K(x) > L$". Let w be the first proof (say, according to the lexicographic order) for a statement of the form "$K(x) > L$". Let z be the integer x such that w proves "$K(x) > L$". It is easy to give a computer program that outputs z : the program enumerates all possible proofs w, one by one, and for the first w that proves a statement of the form "$K(x) > L$", the program outputs x and stops. The length of this program is a $O(1)+\log L$. Thus, if L is large enough, the Kolmogorov complexity of z is less than L. Since w is a proof for "$K(x) > L$" (which is a false statement), we conclude that the theory is inconsistent. Note that the number of computer programs of length L bits is at most $2L + 1$. Hence, for any integer L, there exists an integer $0 \leq x \leq 2L + 1$, such that $K(x) > L$. Thus, for some integer x, the statement "$K(x) > L$" is a true statement that has no proof. □

More recently, Kritchman and Raz [86] used these methods to give a proof of the Second Incompleteness Theorem as well.[10]

This article focuses on algorithmic randomness for infinite objects, but we should mention that there have been many applications of Kolmogorov complexity under the collective title of the *incompressibility method*, based on the observation that algorithmically random strings should exhibit typical behavior for computable processes. For example, as well as the proof of the Incompleteness Theorem above, this method can be used to give average running times for sorting, by showing that if the outcome is not what we would expect then we can compress a random input. See Li and Vitányi [90, Chapter 6] for applications of this technique to areas as diverse as combinatorics, formal languages, compact routing, and circuit complexity, among others. Many results originally proved using Shannon entropy or related methods also have proofs using Kolmogorov complexity[11]. For example, Messner and Thierauf [99] gave a constructive proof of the Lovász Local Lemma using Kolmogorov complexity.

Other applications come from the observation that in some sense Kolmogorov complexity provides an "absolute" measure of the intrinsic complexity of a string. The key is again the notion of conditional Kolmogorov complexity $C(\sigma \mid \tau)$. Then, for example, $C(\sigma \mid \sigma) = O(1)$, and σ is "independent of τ" if $C(\sigma \mid \tau) = C(\sigma) - O(1)$. Researchers comparing two sequences σ, τ representing, say, two

[10] Other recent work has explored the effect of adding axioms asserting the incompressibility of certain strings in a probabilistic way. Bienvenu, Romashchenko, Shen, Taveneaux, and Vermeeren [23] have shown that this kind of procedure does not help to prove new interesting theorems, but that the situation changes if we take into account the sizes of the proofs: randomly chosen axioms (in a sense made precise in their paper) can help to make proofs much shorter under the reasonable complexity-theoretic assumption that NP $\neq$ PSPACE.

[11] Shannon Entropy is more or less an average Kolmogorov Complexity. Hammer et. al. [72] looked at how many inequalities for these concepts are interchangeable.

DNA sequences, or two phylogenetic trees, or two languages, or two pieces of music, have invented many distance metrics, such as the maximum parsimony distance on phylogenetic trees, but it is also natural to use a content-neutral measure of "information distance" like $\max\{C(\sigma \mid \tau), C(\tau \mid \sigma)\}$. There have been some attempts to make this work in practice for solving classification problems, though results have so far been mixed. Of course, C is not computable, but it can be replaced in applications by measures derived from practical compression algorithms. See [90, Sections 8.3 and 8.4]. Also see Bennett et. al. [17] for a survey of these ideas in abstract *Information Distance*. We will not give more details as this area would need a complete survey to itself.

As we will see below, a more recent line of research has used notions of effective dimension based on partial randomness to give new proofs of classical theorems in ergodic theory and obtain new results in geometric measure theory.

4 Some interactions with computability

4.1 Halting probabilities

A first question we might ask is how to generate "natural" examples of algorithmically random reals. A classic example is Chaitin's halting probability. Let U be a universal prefix-free machine and let

$$\Omega = \sum_{U(\sigma)\downarrow} 2^{-|\sigma|}.$$

This number is the measure of the set of sequences X such that U halts on some initial segment of X, which we can interpret as the halting probability of U.

Theorem 4.1 (Chaitin [35]). *Ω above is ML-random.*

Proof. (sketch) Using the Recursion Theorem we will build a prefix-free machine M which as index e within the universal machine U, and e is known in advance. Then we monitor $\Omega_s = \sum_{U(\sigma)\downarrow} 2^{-|\sigma|}[s]$, the stage s approximation. If we see some σ of length $< n - e - 2$ enter Ω_{s+1} with $U(\sigma) = \Omega_s \upharpoonright n$, then we put σ into M_{s+1} causing $1^0 0\sigma$ to enter $\Omega - \Omega_s$, and hence $\Omega_s \upharpoonright n \neq \Omega \upharpoonright n$. (This proof is from [47] and resembles the proof of the unsolvability of the Halting Problem.) □

For any prefix-free machine M in place of U we can similarly define a halting probability. In some ways, halting probabilities are the analogs of computably enumerable sets in the theory of algorithmic randomness. But, as described above, they are left-c.e. in the place of c.e.. Calude, Hertling, Khoussainov, and Wang

[32] showed that every left-c.e. real is the halting probability of some prefix-free machine.

We should perhaps write Ω_U instead of Ω, to stress the dependence of its particular value on the choice of universal machine, but the fundamental properties of Ω do not depend on this choice, much as those of the halting problem do not depend on the specific choice of enumeration of Turing machines. For partial computable functions, the same could also be said. That is $\emptyset' = \{e \mid \varphi_e(e)\downarrow\}$ depends on the precise enumerations of the partial computable functions. But, of course, Myhill proved that up to m-reduction (indeed up to computable permutations) all versions of K had the same degree. So in this sense there is "only one" halting set.

In our case our halting probabilties are reals, and hence what the analogous result would entail would be a continuous version of m-reducibility. (The following is a slight generalization of the earlier version of this concept. See [47])

Definition 4.2 (Solovay [125])**.** We say that $X \leq_S Y$ iff there is a partial computable function f and a constant c such that for any rational $q < Y$ $f(q)\downarrow < X$ and $c(Y-q) \geq (X - f(q))$.

Using this we can effectively convert any Cauchy sequence for Y into one for X which converges just as fast. It is not hard to show that $X \leq_S Y$ means that for all n, $K(X \upharpoonright n) \leq^+ K(Y \upharpoonright n)$. That is because if I know $Y \upharpoonright n$, I can apply f to get $f(Y \upharpoonright n)$ will be within $2^{-(n-\log c)}$ of $X \upharpoonright n$, and hence if we enumerate a small diameter of strings around $f(Y \upharpoonright n)$ we will know that $X \upharpoonright n$ is one of them; and this constant is independent of n. Hence by Schnorr's Theorem, if X is ML-random and $X \leq_S Y$ then Y is ML-random also. Kučera and Slaman [88] showed that every left-c.e. real is reducible to every Ω_U up to Solovay reducibility, and hence all such Ω_U's are equivalent modulo this notion.

Theorem 4.3 (Kučera and Slaman [88])**.** *If X is left-c.e. then $X \leq_S \Omega$. That is Ω is Solovay complete.*

Proof. (Sketch) The proof is an illustrative "measure recycling" one. Given $X = \lim_s X_s$ if $X_{s+1} \upharpoonright n \neq X_s \upharpoonright n$, then the opponent has spent at least 2^{-n} to make this change. We would like Ω_{s+1} to make similar change of around $2^{-(n+c)}$ with c given by the recursion theorem. One way is to put a potential ML-test around $[\Omega_s \upharpoonright n]$ or alternatively issue a description $U(\tau) = \Omega_s \upharpoonright n$, where $|\tau| = n + c$ (meaning, again we build M coded in U by c). The total cost is ≤ 1 so we succeed in forcing Ω to change almost always. □

Hence all universal halting probabilities, being left-c.e. reals are essentially the same. Solovay reducibility is just one measure of relative randomness which can be used, but the reader will see that it is really quite natural. As an analogy to the study of c.e. sets under m-reducibility, we can study the structure of the left-c.e. reals under Solovay reducibility. The following result gives some insight into this structure.

Theorem 4.4 (Downey, Hirschfeldt and Nies [51])**.**

1. *The Solovay degrees of left-c.e. reals forms a distributive dense upper semilattice.*
2. *The join operation is induced by $+$. That is $[\alpha] \vee [\beta] = [\alpha + \beta]$, where $[X]$ denotes the Solovay degree of X.*
3. *The Solovay degree of Ω is join-irreducible That is, if $[\alpha] \vee [\beta] = [\Omega]$ then one of α or β is ML-random. (Also obtained earlier by Demuth [42].)*
4. *Every incomplete Solovay degree of a left c.e. real splits over all lesser ones.*

Recently Barmpalias and Lewis-Pye [10] and Miller [102] proved some fascinating results showing that there is a calculus operating here. Is α is left-c.e. then there is a computable sequence $\alpha_s \leq \alpha_{s+1} \to \alpha$. Then Barmpalias and Lewis-Pye showed the following:

Theorem 4.5 (Barmpalias and Lewis-Pye [10])**.**

1. *If α and β are ML-random left-c.e. reals, then*

$$\frac{\partial \alpha}{\partial \beta} = \lim_s \frac{\alpha - \alpha_s}{\beta - \beta_s}$$

exists, and this is independent of choice of approximations for α and β.

2. *Furthermore $\alpha - \beta$ is ML-random iff $\frac{\partial \alpha}{\partial \beta} \neq 1$.*
3. *And that $\frac{\partial \alpha}{\partial \beta} = \sup\{c \in \mathbb{Q} \mid \alpha - c\beta$ is left-c.e$\} = \inf\{c \in \mathbb{Q} \mid \alpha - c\beta\}$ is right c.e.$\}$.* ∎

Miller [103] extended these results to what are called d.c.e. reals, being those of the form $X - Y$ with X and Y left-c.e. reals. Using this he showed that the nonrandom left-c.e. reals form a real closed field, and ∂ is derivation on this field; meaning it satisfied Leibnitz' Law:

$$\partial(\alpha\beta) = \alpha\partial\beta + \beta\partial\alpha.$$

Consequences of these results are still under active exploration.

Left-c.e. and right-c.e. reals (those of the form $1 - \alpha$ for a left-c.e. α) occur naturally in mathematics. Braverman and Yampolsky [29] showed that they arise in connection with Julia sets, and there is a striking example in symbolic dynamics: A d-dimensional *subshift of finite type* is a certain kind of collection of A-colorings of $\mathbb{Z}^d$, where A is a finite set, defined by local rules (basically saying that certain coloring patterns are illegal) invariant under the shift action

$$(S^g x)(h) = x(h+g) \text{ for } g, h \in \mathbb{Z}^d \text{ and } x \in A^{\mathbb{Z}^d}.$$

Its *(topological) entropy* is an important invariant measuring the asymptotic growth in the number of legal colorings of finite regions. It has been known for some time that entropies of subshifts of finite type for dimensions $d \geqslant 2$ are in general not computable, but the following result gives a precise characterization.

Theorem 4.6 (Hochman and Meyerovitch [75]). *The values of entropies of subshifts of finite type over $\mathbb{Z}^d$ for $d \geqslant 2$ are exactly the nonnegative right-c.e. reals.*

4.2 Algorithmic randomness and relative computability

Solovay reducibility is stronger than Turing reducibility, so Ω can compute the halting problem $\emptyset'$. Indeed Ω and $\emptyset'$ are Turing equivalent, and in fact Ω can be seen as a "highly compressed" version of $\emptyset'$. Other computability-theoretically powerful ML-random sequences can be obtained from the following remarkable result.

Theorem 4.7 (Gács [61], Kučera)[87]). *For every X there is an ML-random Y such that $X \leqslant_{\mathrm{T}} Y$.*

The proof of Theorem 4.7 uses a certain kind of weak-truth table procedure and a kind of "block coding". Recently Barmpalias and Lewis-Pye [11] have established a number of results giving a precise classification of how tightly an arbitrary X can be coded in to a random Y. This has resulted in a completely new optimal coding technique which should have other applications.

Theorem 4.7 and the Turing equivalence of Ω with $\emptyset'$ do not seem to accord with our intuition that random sets should have low "useful information". This phenomenon can be explained by results showing that, for certain purposes, the benchmark set by ML-randomness is too low. A set A has *PA degree* if it can compute a $\{0,1\}$-valued function f with $f(n) \neq \Phi_n(n)$ for all n. (The reason for the name is that this property is equivalent to being able to compute a completion of Peano Arithmetic.) If we can compute a function of this type which is not necessarily $\{0,1\}$ valued, we say that A had *DNC*, *diagonally noncomputable* *degree*. Such a function can be seen as a weak version of the halting problem, but while $\emptyset'$ has PA degree, there are sets of PA degree that are low, in the sense of Section 2.2, and hence are far less powerful than $\emptyset'$.

Theorem 4.8 (Stephan [126])**.** *If an ML-random sequence has PA degree then it computes $\emptyset'$.*

Thus there are two kinds of ML-random sequences. Ones that are complicated enough to somehow "simulate" randomness, and "truly random" ones that are much weaker.

On the other hand, if we remove the restriction that f must be $\{0,1\}$-valued we get the following:

Theorem 4.9 (Kučera [87])**.** *If A is ML-random then A computes a DNC function.*

The explanation of the apparent paradox is that if a function is $\{0,1\}$-valued, then saying it does not have value 1, means it must have value 0, and conversely. If more values are possible, saying what it is *not* does not imply what it is.

Proof. Let A be ML-random. Let $f(n)$ be the position of $A \upharpoonright n$ in some effective listing of finite binary strings. Since A is ML-random,

$$K(f(n)) =^{+} K(A \upharpoonright n) \geq^{+} n,$$

by Schnorr's Theorem, Theorem 2.9. On the other hand, if $varphi_n(n) \downarrow$, then $K(\varphi_n(n)) \leq^{+} n$, so there are only finitely many n such that $f(n) = \varphi_n(n)$. By altering f at these finitely many places, we obtain an A-computable DNC function. □

Kučera's Theorem above received a lot of attention and generalizations. Crucial is the use of Schnorr's Theorem; but this also has generalizations. For example, we call a set A *complex* if there is an order h such that $K(A \upharpoonright h(n)) \geq n$ for some computable order h, and we say that A is *autocomplex* if A is h-complex for some A-computable order.

Theorem 4.10. *(Kjos-Hanssen, Merkle, and Stephan [82]). A set is autocomplex iff it is of DNC degree.*

For more on this story, see [47], Ch. 8.

4.2.1 Stronger randomness

It is known that the class of sequences that can compute $\emptyset'$ has measure 0, so almost all ML-random sequences are in the second class. This fact is a special case of the following classical theorem

Theorem 4.11 (de Leeuw, et. al. [41], Sacks [119]). *Suppose that A is noncomputable. Then $\lambda\{X \mid A \leq_T X\} = 0$.*

Proof. This is included as it is a classic "majority vote" argument. Suppose that

$$\lambda\{X \mid A \leq_T X\} > 0.$$

Then for some fixed Φ, $\lambda\{X \mid \Phi^X = A\} > 0$, by the fact that there are only a countable number of procedures Φ. Then by Lebesgue Density, we can work in some cone of reals C where the relative density

$$\frac{\lambda\{X \mid \Phi^X = A \wedge \sigma \prec X\}}{\lambda\{Y \mid \sigma \prec Y\}} > \frac{3}{4}.$$

Then to compute $A \restriction n$ for $n \geq |\sigma|$ wait till $\frac{3}{4}$ of the oracles X extending σ have $\Phi^X \restriction\downarrow$ with a common value. This correctly computes $A \restriction n$. □

One way to ensure that a sequence is in that class of reals with little usable information is to increase the complexity of our tests by relativizing them to noncomputable oracles. It turns out that iterates of the Turing jump are particularly natural oracles to use. Let $\emptyset^{(0)} = \emptyset$ and $\emptyset^{(n+1)} = (\emptyset^{(n)})'$. We say that X is *n-random* if it passes all Martin-Löf tests relativized to to $\emptyset^{(n-1)}$. Thus the 1-random sequences are just the ML-random ones, while the 2-random ones are the ones that are ML-random relative to the halting problem. These sequences have low computational power in several ways. For instance, they cannot compute any noncomputable c.e. set, and in fact the following holds.

Theorem 4.12 (Kurtz [89]). *If X is 2-random and Y is computable relative both to $\emptyset'$ and to X, then Y is computable.*

A precise relationship between tests and the dichotomy mentioned above was established by Franklin and Ng [58] using another more technical notion of randomness.

We remark that of course, we can relativise the notion of Kolmogorov complexity, and see that as an analog of Schnorr's Theorem we have X is n-random iff $K^{\emptyset^{(n-1)}}(X \restriction k) \geq^+ k$ for all k.

Quite remarkably, there is interdefinability between C and $C^{\emptyset^{(n)}}$ and similarly K and $K^{\emptyset^{(n)}}$. We will need the notion of notion of conditional Kolmogorov complexity $C(\sigma \mid \tau)$ of a string σ given another string τ (and same for K).

Theorem 4.13 (Vereshchagin [129])**.** $C^{\emptyset'}(\sigma) =^+ \limsup_n C(\sigma|n)$.

Proof. Let M be our fixed universal plain machine. Let $M(\tau, n) = M^{\emptyset'}(\tau)[n]$, where both the oracle and the universal machine are being approximated. If n is large enough and τ is a minimal-length $M^{\emptyset'}$-program for σ, then $M(\tau, n) = \sigma$, whence $C(\sigma \mid n) \upharpoonright |\tau| =^+ C^{\emptyset'}(\sigma) + O(1)$. Thus $\limsup_n C(\sigma|n) \leq C^{\emptyset'}(\sigma)$.

For the other direction, let $k = \limsup_n C(\sigma|n) + 1$. Let $V_n = \{\sigma \mid C(\sigma|n) < k$ We have $|V_n| < 2k$ for all n. Let $B = \{\tau \mid \exists m \forall n \geq m(|V_n \cup \{\tau\}) < 2^k\}$. Then B is $\emptyset'$-c.e. (uniformly in k), $\sigma \in B$, and $|B| < 2k$. Since we can describe σ from $\emptyset'$ by giving its position in the enumeration of B as a string of length k, we have $C^{\emptyset'}(\sigma) \leq^+ k =^+ \limsup_n C(\sigma|n)$. □

Slightly more complex methods yield the following.

Theorem 4.14 (Bienvenu, Muchnik, Shen, and Vereschagin [21])**.** $K^{\emptyset'}(\sigma) =^+ \limsup_n K(\sigma|n)$.

It follows that, for example, n-randomness can be defined using K alone, which is surprising in that on the face of it $K^{\emptyset'}$ would seem unrelated to K. We remark that earlier Solovay [125] has looked at relationships between K and $K^{\emptyset'}$ and some of this material can be found in [47]. Also there had been earlier results showing that 2-randomness, in particular, is naturally definable. We have seen that Martin-Löf showed that for no X is $C(X \upharpoonright n) \geq^+ n$ for all n. But it is possible that $C(X \upharpoonright n) \geq^+ n$ for *infinitely many* n, as shown by Martin-Löf [96], and he also showed that such sets were ML-random.

Theorem 4.15 (Miller [100], Nies, Stephan and Terwijn [112])**.** *X is 2-random iff $\exists^\infty n C(X \upharpoonright n) \geq^+ n$.*

The proof of this result in [112] uses an interesting notion called a compression function. We can also look at the K-analog. The maximal complexity n can be is $n + K(n) + O(1)$, as proved by Solovay [125]. Yu, Ding and Downey [134] proved that if a set is 3-random then it infinitely often will reach this complexity on its initial segments of length n. The final piece of the puzzle was supplied by Joe Miller:

Theorem 4.16 (Miller [101])**.** *X is 2-random iff $\exists^\infty n K(X \upharpoonright n) \geq^+ n + K(n)$.*

4.2.2 Computational depth

In general, among ML-random sequences, computational power (or “useful information”) is inversely proportional to level of randomness. The following is one of many results attesting to this heuristic.

Theorem 4.17 (Miller and Yu [104])**.** *Let $X \leqslant_{\mathrm{T}} Y$. If X is ML-random and Y is n-random, then X is also n-random.*

Thus the notion that a random sequence has a "high information content" seems quite wrong. What is missing is the word "useful". There might be a lot of information but it is not accessible to a computational procedure. So when does a real contain useful in formation? One line of investigation in this area was pioneered by Bennett who defined X to be K-deep if we cannot know about A in any computable time. To wit, let K^t be a time bounded version of prefix-free complexity.

Definition 4.18. X is called *Bennett deep* or simply deep for short if for all all computable t and for all c and almost all n,

$$K^t(X \upharpoonright n) - K(X \upharpoonright n) > c.$$

If X is not deep it is called *shallow.*

The intuition is that X is deep because it contains a lot of information which is difficult to discover.

Theorem 4.19 (Bennett [18])**.**

1. *All computable and ML-random sets are shallow.*
2. *There are deep c.e. sets, such as the halting problem.*

The notion of depth has proven quite fruitful in giving insight into intrinsic information in languages, and several further variations on the notion, mainly involving orders[12] (in place of c) and C in place of K) have been studied. For example, Moser and Stephan [106] showed that a degree is high iff it contains a "strongly" deep set. See, for instance, [6, 7, 106], etc. As Moser [105] showed, all of these notions have a common interpretation in terms of computable time bounds and compression ratios.

All of this might lead one to suspect that Ω is a bit player in the area of algorithmic randomness. But perhaps the following two theorems say that in some sense it is a central concept.

Theorem 4.20 (Downey, Hirschfeldt, Miller, Nies [50])**.** *If X is 2-random then there is a set Y such that $X = \Omega^Y$. That is, almost all random reals are Ω-numbers relative to an oracle.*

[12] That is, a computable nondecreasing unbounded function.

In the same was that Ω has c.e. degree we get the following.

Theorem 4.21 (Kurtz [89]). *If X is 2-random then there is a $Z <_T X$ such that $deg_T(X)$ is c.e. relative to Z.*

The original proof of this theorem uses a technique called "*measure risking*" which allows for a procedure in a construction to be undefined on some small measure part of 2^ω, and after the fact, it is argued that the construction succeeds on all 2-randoms. Recently this idea has been portrayed using what has been dubbed by Shen to be a "fireworks" argument. We will illustrate this new method with a slightly easier proof that every 2-random bounds a 1-generic degree, where Z is called 1-generic iff for all c.e. sets of strings W, either $\exists \sigma \prec Z$ and σ has not extension in W, or there is some $\sigma \prec Z$ and $\sigma \in W$.

The fireworks metaphor is the following. We wish to purchase some fireworks from a seller who claims that all of them are good, but perhaps we are using them for an important party and it is crucial that they work when at the time. We have a lot of money so can test a large number. What we do is to ask the seller to show us, say, 100. We pick a number n between 1 and 100, randomly. We then test the first $n-1$ of the fireworks and is any fail then we reject the seller's package. If all work we accept and will use the n-th one for the party. (Of course we will need to pay for the first n, but that is another story.) For the seller to have sold us a dud, they would have to have guessed our n. The probability is at most $\frac{1}{100}$.

In computability, we use this idea for probabilistic forcing. Imagine we are meeting the requirements R_e asking that either we have some $\sigma \in W_e$ with $\sigma \prec Z$ or want to show that for some $\sigma \prec Z$, and $\tau \in W_e$ $\sigma \not\preceq \tau$.

Typically this would be done using the finite extension method, which is Cohen forcing with conditions being finite strings. Thus $\emptyset'$ can carry out such a construction. Here we need a probabilistic argument which will work for a 2-random oracle. Now in this construction we only need to do two things. If we are in a situation that we have build Z_s and there is no extension in W_e then we win by luck; the "passive guess." So, what we will do is begin a step by step construction which, for a fixed e, would pick a random $n(e)$ in some suitable interval (depending on the priority) and work with the passive guess for $n(e)$ many e-steps. That is, we assume that our guess is correct, but sometime later if we find out that it was not at that stage (i.e. Z_s actually had an extension in W_e) we would make another step. If we run out of steps say at s_1, then we will stop the construction doing the active guess seeking some $\tau \in W_e$ extending Z_{s_1}.

The random oracle is the one supplying the answers, and Γ^R would build a Martin-Löf type test, here a $\emptyset'$-computable one, such that oracles can only fail on such tests, in the sense that the construction would get stuck only inside open sets generated by the tests. For an $\emptyset'$-computable one, the construction would succeed outside it and hence a 2-random could carry the construction out. We refer the reader to Bienvenu and Patey [22] for more details and an interesting application to "bushy tree" forcing.

Of course it is impossible to combine Theorems 4.20 and 4.21. We remark that Ω showcases the difference between being c.e. relative to some set and "CEA" (c.e. relative to and *above*). In fact relativization of Ω is somewhat counter-intuitive:

Theorem 4.22 (Downey, Hirschfeldt, Miller, Nies [50]). *There are sets $A =^* B$ (i.e. the symmetric difference is finite) such that $\Omega^B |_T \Omega^A$. In fact, they are relatively random.*

For recent related work on Ω as an operator see Hölzl et. al. [76].

4.2.3 Calibrating randomness

There are many other interesting calibrations of algorithmic randomness. As we have seen, Schnorr [120] argued that his left-c.e. martingale, Theorem 2.4, characterization of ML-randomness shows that this is an intrinsically *computably enumerable* rather than *computable* notion. As well as defining computable randomness he also defined a concept now called *Schnorr randomness*, which is like the notion of computable randomness mentioned below Definition 2.3 but with an extra effectiveness condition on the rate of success of martingales. He also showed that X is Schnorr random iff it passes all Martin-Löf tests $T_0, T_1, \ldots$ such that the measures $\lambda(T_n)$ are uniformly computable (i.e., the function $n \mapsto \lambda(T_n)$ is computable in the sense of Section 5.4 below). It follows immediately from their definitions in terms of martingales that ML-randomness implies computable randomness, which in turn implies Schnorr randomness. It is more difficult to prove that none of these implications can be reversed. In fact, these levels of randomness are close enough that they agree for sets that are somewhat close to computable, as shown by the following result, where highness is as defined in Section 2.2.

Theorem 4.23 (Nies, Stephan, and Terwijn [112]). *Every high Turing degree contains a set that is computably random but not ML-random and a set that is Schnorr random but not computably random. This fact is tight, however, because every nonhigh Schnorr random set is ML-random.*

One last example of a variation is not called *Kurtz random* and is defined in a slightly different way.

Definition 4.24. We say that X is *Kurtz random* iff for all c.e. open sets O of measure 1, $X \in O$. The complement of O is called a Kurtz (null) test.

Kurtz randomness is a weak notion of randomness but coincides with ML-randomness on the hyperimmune-free degrees[13] We mention Kurtz randomness because it actually comes up in classifying theorems and randomness. For example, it can be shown that (with the correct definitions) a suitably computable bounded function on a closed interval has a Reimann integral iff it is undefined on Kurtz test. As we will discuss, various notions of algorithmic randomness arise naturally in applications. We already mentioned the fact that Kurtz randomness arises in the study of random (divergent) Fourier series [45]. Schnorr randomness arises a there also for convergent series.

4.3 Randomness-theoretic weakness

As mentioned above, X is ML-random iff $K(X \upharpoonright n) \geqslant n - O(1)$, i.e., X's initial segments have very high complexity. There are similar characterizations of other notions of algorithmic randomness, as well as of notions arising in other parts of computability theory, in terms of high initial segment complexity. But what if the initial segments of a sequence have *low* complexity? Such sequences have played an important role in the theory of algorithmic randomness, beginning with the following information-theoretic characterization of computability.

Theorem 4.25 (Chaitin [35]). *$C(X \upharpoonright n) \leqslant^+ C(n)$ iff X is computable.*

Proof. (Sketch) The first part of this proof is to use a combinatorial pigeonhole argument to show that

$$|\{\sigma \in 2^n \mid C(\sigma) < n + d\}| \leq O(2^d).$$

That is, only few strings have short descriptions, *and this is independent of n.* Now between lengths 2^k and 2^{k+1} there will be C-random lengths where $C(n) = \log |n|$. We can use this fact to build a computable tree of width 2^d such that if $C(A \upharpoonright n) \leq C(n) + d$ for all n, then A is a member of this tree, and hence is computable. □

It is also true that if X is computable then $K(X \upharpoonright n) \leqslant K(n) + O(1)$. Chaitin [36] considered sequences with this property, which are now called *K-trivial.* He showed that every K-trivial sequence is $\emptyset'$-computable[14], and asked whether they

[13] A has hyperimmune-free degree iff for all functions $f \leq_T A$, there is a computable function g such that for all x, $f(x) < g(x)$. Some authors have called these *computably dominated* for obvious reasons. If A does not have hyperimmune-free degree it is said to have hyperimmune degree. All degrees computable from the halting problem are hyperimmune and the non-zero ones contain Kurtz random reals.

[14] This uses a similar argument to that of Theorem 4.25, but now note that we cannot know $K(n)$ for random n, in the same way that $C(n)$ will be $\log |n|$. Only $\emptyset'$ can know this.

are all in fact computable. Solovay [125] answered this question by constructing a noncomputable K-trivial sequence. Here is a simple construction of a K-trivial noncomputable real.

Theorem 4.26 (Zambella [135], after Solovay [125]). *There is a c.e. noncomputable K-trivial set.*

Proof. This proof is taken from Downey, Hirschfeldt, Nies and Stephan [52]. We define a c.e. set A as follows:

$$A_{s+1} = A_s \cup \{x \mid x \in W_{e,s} \wedge e \text{ least with } W_{e,s} \cap A_s = \emptyset \wedge \sum_{s \geq y \geq x} 2^{-K_s(y)} < 2^{-(2e+1)}\}.$$

Then A is K-trivial because we can build a machine M such that for all n, $K_M(A \upharpoonright n) \leq K(n) + 1$. We can copy $U(n)[s]$ with σ describing $n \leq s$ by using 1σ, unless we change $A_s \upharpoonright n$ in which case we use a string $10^e\tau$ with τ of length $\geq e + 1$. The overall cost of the extra material is ≤ 1 and hence there is enough space in M to build the extra strings. □

The construction above is now in a class of "cost function" constructions which are summarized by "do what is cheap enough." In [52] it is shown that K-trivials are in fact characterized by cost functions. The class of K-trivials has several remarkable properties. It is a naturally definable *countable* class, contained in the class of low sets (as defined in Section 2.2, where we identify a set with its characteristic function, thought of as a sequence), but with stronger closure properties. (In technical terms, it is what is known as a *Turing ideal.*) Post's problem asked whether there are computably enumerable sets that are neither computable nor Turing equivalent to the halting problem. Its solution in the 1950's by Friedberg and Muchnik introduced the somewhat complex priority method, which has played a central technical role in computability theory since then. Downey, Hirschfeldt, Nies, and Stephan [52] showed that K-triviality can be used to give a simple priority-free solution to Post's problem.

Most significantly, there are many natural notions of randomness-theoretic weakness that turn out to be equivalent to K-triviality.

Theorem 4.27 (Nies [108], Nies and Hirschfeldt for (1) → (3)). *The following are equivalent.*

1. *A is K-trivial.*

2. *A is computable relative to some c.e. K-trivial set.*

3. *A is* low for K, *meaning that A has no compression power as an oracle. i.e., that $K^A(\sigma) \geqslant K(\sigma) - O(1)$, where K^A is the relativization of prefix-free Kolmogorov complexity to A.*

4. *A is* low for ML-randomness, *meaning that A does not have any derandomization power as an oracle, i.e., any ML-random set remains ML-random when this notion is relativized to A.*

There are now over a dozen other characterizations of K-triviality. Some appear in [47, 109], and several others have emerged more recently (e.g. [64]). These have been used to solve several problems in algorithmic randomness and related areas.

Theorem 4.27 (2) above says that K-trivials are computable from c.e. K-trivials. In some sense this means that that they are intrinsically c.e. and cannot, it seems, by e.g. a forcing construction. Focussing on cost functions has allowed for a number of recent advances in the area. Nies [110] (which was available for some years before it was submitted) was the first to realize that this was a powerful abstraction in the area. Some recent examples of applications include [70, 65, 64]. This material seems tied up with derandomization power via another reducibility $A \leq_{LR} B$ meaning that $ML^A \supseteq ML^B$: everything A derandomized, B does too.

Lowness classes have also been found for other randomness notions. For Schnorr randomness, for instance, lowness can be characterized using notions of traceability related to concepts in set theory, as first explored by Terwijn and Zambella [127].

For example, we have the following.

Theorem 4.28 (Terwijn and Zambella [127]). *X is low for Schnorr tests iff X is* computably traceable. *This means that there is a computable order h such that for all $f \leq_T X$, we can compute an array of canonical finite sets $\{D_{g(n)} \mid n \in \mathbb{N}\}$ called a* trace *such that for all n, $f(n) \in D_{g(n)}$.*

Finally, using some earlier work of Bendregal and Nies, we have the following for the randomness concept:

Theorem 4.29 (Kjos-Hanssen, Nies and Stephan [83]). *X is low for Schnorr randomness iff X is low for Schnorr tests iff X is computably traceable.*

It is easy to see that if X is computably traceable it must be hyperimmune-free. And it is not hard to prove that there are continuum many computably traceable sets. The K-trivials are a countable collection of low sets below $\emptyset'$. Thus the classes are very different. In particular, it is not possible to define

Schnorr randomness using K, nor ML-randomness using Schnorr tests, with any relativizable definition.

Similar characterizations were found for lowness for Kurtz randomness. Building on work of several authors, the final characterization for Kurtz randomness was the following.

Theorem 4.30 (Greenberg and Miller [66]). *A is low for Kurtz randomness iff A is hyperimmune-free and not DNC.*

Nies [108] showed that if X is low for computable randomness then X is computable.

Some of this work is related to coarse and generic computability mentioned below.

We remark that the use of traceing has become quite influential in computability theory. Can we find find a combinatorial definition of K-triviality, (i.e. not involving K) Nies and others suggested that it was related to *jump*-traceability. Let J^X denote the universal partial computable function. That is $J^X(e)$ denotes the actual value[15] (if any) of $\Phi_e(e)$.

Definition 4.31 (Figueira, Stephan and Nies [55]). If h is an order, we say that A is *jump traceable* at order h, if, for any A-partial computable function p, there is computable collection of $\{T_e \mid e \in \mathbb{N}\}$ of c.e. sets such that $p^A(e) \in T_e$ and $|T_e| \leq h(e)$ for all e.

We say that A is *strongly jump traceable* iff it is jump traceable for all orders h, iff J^A is jump traceable for all orders h.

For example, a c.e. set A is superlow (meaning $A' \equiv_{tt} \emptyset'$) iff A is jump traceable. In [55] it is shown that at order $h(n) = 2^{2n}$ there are $2^{\aleph_0}$ many h-jt sets.

Theorem 4.32 (Cholak, Downey and Greenberg [37]). *There is an order h where A being h-jt implies A is K-trivial.*

The order from [37] is around $\log\log n$, but this is certainly not optimal.

Question 4.33. Is there an order-characterization of being K-trivial. The guess would be that it would not involve a single order but a collection of a certain type, something like if $\{h_e \mid h_e \in S\}$ has property X (like some sum coverges) then A is K-trivial iff A is h_e-jump traceable for all $h_e \in S$.

Now it might seem that strong jump traceability is an artifact of stidies into randomness and now directly related but several results show that this is not the case. For example:

[15] That is not just if it halts or not.

Theorem 4.34 (Greenberg, Hirschfeldt, Nies [63]). *A is sjt iff A is computable from every superlow ML-random.*

There are similar results about ML-random reals X which are superhigh, which means that $X' \equiv_{tt} \emptyset''$ (see [63]), and characterizations involving another notion of randomness called *Demuth randomness*, which is defined via generalized ML-tests. We also remark that super jump traceable sets have been used to solve open questions in classical computability. For instance,we say that a c.e. set W is a cea-operator if for all Y, $Y <_T W^Y$. A classical theorem of Jockusch and Shore shows that for all such W there is a c.e. set Y with $Y \oplus W^Y \equiv_T \emptyset'$. This is called *pseudo-jump inversion.* Downey and Greenberg solved a longstanding question above cone avoidance by taking W to be the construction of a noncomputable strongly jump traceable set (so that W^Y was "very high" in that $\emptyset'$ would be stringly jump traceable relative to W^Y).

Theorem 4.35 (Downey and Greenberg [44]). *There is a noncomputable c.e. set B computable from all c.e. sets C with $\emptyset'$ strongly jump traceable relative to C. That is, W is a c.e. operator which cannot avoid upper cones under inversion.*

Recent unpublished work of Downey, Greenberg and Turetsky shows that B can be chosen to be superhigh. For the latest word here see Greenberg and Turetsky [71] for a survey of results and techniques.

5 Some applications

5.1 Incompressibility and information content

This article focuses on algorithmic randomness for infinite objects, but we should mention that there have been many applications of Kolmogorov complexity under the collective title of the *incompressibility method*, based on the observation that algorithmically random strings should exhibit typical behavior for computable processes. For example, this method can be used to give average running times for sorting, by showing that if the outcome is not what we would expect then we can compress a random input. See Li and Vitányi [90, Chapter 6] for applications of this technique to areas as diverse as combinatorics, formal languages, compact routing, and circuit complexity, among others. Many results originally proved using Shannon entropy or related methods also have proofs using Kolmogorov complexity. For example, Messner and Thierauf [99] gave a constructive proof of the Lovász Local Lemma using Kolmogorov complexity.

Other applications come from the observation that in some sense Kolmogorov complexity provides an "absolute" measure of the intrinsic complexity of a string.

We can define a notion of conditional Kolmogorov complexity $C(\sigma \mid \tau)$ of a string σ given another string τ. Then, for example, $C(\sigma \mid \sigma) = O(1)$, and σ is "independent of τ" if $C(\sigma \mid \tau) = C(\sigma) - O(1)$. Researchers comparing two sequences σ, τ representing, say, two DNA sequences, or two phylogenetic trees, or two languages, or two pieces of music, have invented many distance metrics, such as the maximum parsimony distance on phylogenetic trees, but it is also natural to use a content-neutral measure of "information distance" like $\max\{C(\sigma \mid \tau), C(\tau \mid \sigma)\}$. There have been some attempts to make this work in practice for solving classification problems, though results have so far been mixed. Of course, C is not computable, but it can be replaced in applications by measures derived from practical compression algorithms. See [90, Sections 8.3 and 8.4].

5.2 Effective dimensions

If $X = x_0 x_1 \dots$ is random, then we might expect a sequence such as $x_0 00 x_1 00 x_2 00 \dots$ to be "$\frac{1}{3}$-random". Making precise sense of the idea of partial algorithmic randomness has led to significant applications. Hausdorff used work of Carathéodory on s-dimensional measures to generalize the notion of dimension to possibly nonintegral values, leading to concepts such as Hausdorff dimension and packing dimension. Much like algorithmic randomness can make sense of the idea of individual reals being random, notions of partial algorithmic randomness can be used to assign dimensions to individual reals.

The measure-theoretic approach, in which we for instance replace the uniform measure λ on 2^ω by a generalized notion assigning the value $2^{-s|\sigma|}$ to $[\sigma]$ (where $0 < s \leqslant 1$), was translated by Lutz [91, 92] into a notion of *s-gale*, where the fairness condition of a martingale is replaced by $f(\sigma) = 2^{-s}(f(\sigma 0) + f(\sigma 1))$. We can view s-gales as modeling betting in a hostile environment (an idea due to Lutz), where "inflation" is acting so that not winning means that we automatically lose money. Roughly speaking, the effective fractal dimension of a sequence is then determined by the most hostile environment in which we can still make money betting on this sequence.

Mayordomo [97] and Athreya, Hitchcock, Lutz, and Mayordomo [8] found equivalent formulations in terms of Kolmogorov complexity, which we take as definitions. (Here it does not matter whether we use plain or prefix-free Kolmogorov complexity.)

Definition 5.1. [16] Let $X \in 2^\omega$. The *effective Hausdorff dimension* of X is

$$\dim(X) = \liminf_{n\to\infty} \frac{K(X \upharpoonright n)}{n}.$$

The *effective packing dimension* of X is

$$\mathrm{Dim}(X) = \limsup_{n\to\infty} \frac{K(X \upharpoonright n)}{n}.$$

It is not hard to extend these definitions to elements of $\mathbb{R}^n$, yielding effective dimensions between 0 and n. They can also be relativized to any oracle A to obtain the effective Hausdorff and packing dimensions $\dim^A(X)$ and $\mathrm{Dim}^A(X)$ of X relative to A.

It is of course not immediately obvious why these notions are effectivizations of Hausdorff and packing dimension, but crucial evidence of their correctness is provided by *point to set principles*, which allow us to express the dimensions of sets of reals in terms of the effective dimensions of their elements. The most recent and powerful of these is the following, where we denote the classical Hausdorff dimension of $E \subseteq \mathbb{R}^n$ by $\dim_{\mathrm{H}}(E)$, and its classical packing dimension by $\dim_{\mathrm{p}}(E)$.

Theorem 5.2 (Lutz and Lutz [93])**.**

$$\dim_{\mathrm{H}}(E) = \min_{A\subseteq\mathbb{N}} \sup_{X\in E} \dim^A(X).$$

$$\dim_{\mathrm{p}}(E) = \min_{A\subseteq\mathbb{N}} \sup_{X\in E} \mathrm{Dim}^A(X).$$

For certain well-behaved sets E, relativization is actually not needed, and the classical dimension of E is the supremum of the effective dimensions of its points. In the general case, it is of course not immediately clear that the minima mentioned in Theorem 5.2 should exist, but they do. Thus, for example, to prove a lower bound of α for $\dim_{\mathrm{H}}(E)$ it suffices to prove that, for each $\varepsilon > 0$ and each A, the set E contains a point X with $\dim^A(X) > \alpha - \varepsilon$. In several applications, this argument turns out to be easier than ones directly involving classical dimension. This fact is somewhat surprising given the need to relativize to arbitrary oracles, but in practice this issue has so far turned out not to be an obstacle.

For example, Lutz and Stull [95] obtained a new lower bound on the Hausdorff dimension of generalized sets of Furstenberg type; Lutz [94] showed that a fundamental intersection formula, due in the Borel case to Kahane and Mattila, is true for arbitrary sets.

[16] Strictly speaking, this should be viewed as a Theorem, but it has become the standard definition. See [47], Chapter 13 for the full story.

$E \subseteq \mathbb{R}^n$ is called a *Kakeya Set* for every point u on the unit sphere S^{n-1}, there is some point v such that the segment $\{su + v : s \in [0,1]\} \subseteq E$; that is E has unit lines in every direction. This is an important classical concept which has applications from harmonic analysis to extractors in complexity theory. Using the point to set principle, Lutz and Lutz [93] gave a new proof of the two-dimensional case (originally proved by Davies) of the well-known Kakeya conjecture, which states that, for all $n \geqslant 2$, if a subset of $\mathbb{R}^n$ has lines of length 1 in all directions, then it has Hausdorff dimension n. The method used by Lutz and Lutz filtered through the following result.

Theorem 5.3 (Lutz and Lutz [93]). *Let $a, b, x \in \mathbb{R}$. If a is ML-random and x is ML-random relative to the point (a, b), then the effective Hausdorff dimension of the point $(x, ax + b)$ is 2.*

The proof of Theorems 5.2 and 5.3 are familiar types of Kolmogorov complexity calculations, and are far from the classical techniques. Using Theorems 5.2 and 5.3 Lutz and Lutz gave the following proof of Davies theorem.

Proof. Let $E \subseteq \mathbb{R}^2$ be a Kakeya set, and let w be the minimizing oracle of Theorem 5.2. Let a be random ML-relative to w, and let b be such that the intersection of E with the line $y = ax + b$ contains a segment. Choose x random relative to (a, b, w) such that $(x, ax + b) \in E$. Then $\dim(E) = \sup_{z \in E} \dim^w(z) \geq \dim^w(x, ax + b)$, which is 2 by Theorem 5.3, applied relative to the oracle w. □

There had been earlier applications of effective dimension, for instance in symbolic dynamics, whose iterative processes are naturally algorithmic. For example, Simpson [123] generalized a result of Furstenberg as follows. Let A be finite and G be either $\mathbb{N}^d$ or $\mathbb{Z}^d$. A closed set $X \subseteq A^G$ is a *subshift* if it is closed under the shift action of G on A^G (see Section 4.1).

Theorem 5.4 (Simpson [123]). *Let A be finite and G be either $\mathbb{N}^d$ or $\mathbb{Z}^d$. If $X \subseteq A^G$ is a subshift then the topological entropy of X is equal both to its classical Hausdorff dimension and to the supremum of the effective Hausdorff dimensions of its elements.*

In currently unpublished work, Day has used effective methods to give a new proof of the Kolmogorov-Sinai theorem on entropies of Bernoulli shifts.

There are other applications of sequences of high effective dimension, for instance ones involving the interesting class of *shift complex* sequences. While initial segments of ML-random sequences have high Kolmogorov complexity, not all segments of such sequences do. Random sequences must contain arbitrarily long strings of consecutive 0's, for example. For example, if we knew that there was no sequence of 0's of length more than 12, but infinitely many of length 12, we could easily construct a martingales to succeed: wait for 12 0's and bet that the next bit was a 1!

However, for any $\varepsilon > 0$ there are *ε-shift complex* sequences Y such that for any string σ of consecutive bits of Y, we have $K(\sigma) \geqslant (1-\varepsilon)|\sigma| - O(1)$. These sequences can be used to create tilings with properties such as certain kinds of pattern-avoidance, and have found uses in symbolic dynamics. See for instance Durand, Levin, and Shen [53] and Durand, Romashchenko, and Shen [54].

5.3 Randomness amplification

Many practical algorithms use random seeds. For example, the important *Polynomial Identity Testing* (*PIT*) problem takes as input a polynomial $P(x_1, \ldots, x_n)$ with coefficients from a large finite field and determines whether it is identically 0. Many practical problems can be solved using a reduction to this problem. There is a natural fast algorithm to solve it randomly: Take a random sequence of values for the variables. If the polynomial is not 0 on these values, "no" is the correct answer. Otherwise, the probability that the answer is "yes" is very high. It is conjectured that PIT has a polynomial-time deterministic algorithm,[17] but no such algorithm is known.

Thus it is important to have good sources of randomness. Some (including Turing) have believed that randomness can be obtained from physical sources, and there are now commercial devices claiming to do so. At a more theoretical level, we might ask question such as:

1. Can a weak source of randomness always be amplified into a better one?
2. Can we in fact always recover full randomness from partial randomness?
3. Are random sources truly useful as computational resources?

In our context, we can consider precise versions of such questions by taking randomness to mean algorithmic randomness, and taking all reduction processes to be computable ones. One way to interpret the first two questions then is to think of partial randomness as having nonzero effective dimension. For example, for packing dimension, we have the following negative results.

[17] This conjecture comes from the fact that PIT belongs to a complexity class known as BPP, which is widely believed to equal the complexity class P of polynomial-time solvable problems, since, in highly celebrated work, Impagliazzo and Wigderson [78] showed in the late 1990's that if the well-known Satisfiability problem is as hard as generally believed, then indeed BPP = P.

Theorem 5.5 (Downey and Greenberg [43]). *There is an X such that* $\mathrm{Dim}(X) = 1$ *and X computes no ML-random sequence. (This X can be built to be of* minimal degree*, which means that every X-computable set is either computable or has the same Turing degree as X. It is known that such an X cannot compute an ML-random sequence.)*

Theorem 5.6 (Conidis [39])). *There is an X such that* $\mathrm{Dim}(X) > 0$ *and X computes no Y with* $\mathrm{Dim}(Y) = 1$.

On the other hand, we also have the following strong positive result.

Theorem 5.7 (Fortnow, Hitchcock, Pavan, Vinochandran, and Wang [56]). *If* $\varepsilon > 0$ *and* $\mathrm{Dim}(X) > 0$ *then there is an X-computable Y such that* $\mathrm{Dim}(Y) > 1-\varepsilon$. *(In fact, Y can be taken to be equivalent to X via polynomial-time reductions.)*

For effective Hausdorff dimension, the situation is quite different. Typically, the way we obtain an X with $\dim(X) = \frac{1}{2}$, say, is to start with an ML-random sequence and somehow "mess it up", for example by making every other bit a 0. This kind of process is reversible, in the sense that it easy to obtain an X-computable ML-random. However, Miller [102] showed that it is possible to obtain sequences of fractional effective Hausdorff dimension that permit no randomness amplification at all.

Theorem 5.8 (Miller [102]). *There is an X such that* $\dim(X) = \frac{1}{2}$ *and if* $Y \leqslant_{\mathrm{T}} X$ *then* $\dim(Y) \leqslant \frac{1}{2}$.

The proof of Theorem 5.8 is a novel forcing argument resulting in a Δ^0_2 set. The classification of such fractional dimension degrees is completely open.

Theorem 5.8 shows that effective Hausdorff dimension *cannot* in general be amplified. (In this theorem, the specific value $\frac{1}{2}$ is only an example.) Greenberg and Miller [67] also showed that there is an X such that $\dim(X) = 1$ and X does not compute any ML-random sequences.

There is one very intriguing open question here: We know that if A is random and $A_0 \oplus A_1 = A$ then A_i is relatively random to A_{1-i}. This fact is a basic result called van Lambalgen's Theorem. It implies that no random set can have minimal Turing degree.

Question 5.9. Can a set of Effective Hausdorff dimension have minimal Turing degree?

A positive answer would give a proof of a strengthening of the Greenberg-Miller Theorem.

Interestingly, Zimand [136] showed that for *two* sequences X and Y of nonzero effective Hausdorff dimension that are in a certain technical sense sufficiently independent, X and Y together can compute a sequence of effective Hausdorff dimension 1.

In some attractive recent work, it has been shown that there is a sense in which the intuition that every sequence of effective Hausdorff dimension 1 is close to an ML-random sequence is correct. The following is a simplified version of the full statement, which quantifies how much randomness can be extracted at the cost of altering a sequence on a set of density 0. Here $A \subseteq \mathbb{N}$ has *(asymptotic) density* 0 if $\lim_{n\to\infty} \frac{|A\restriction n|}{n} = 0$.

Theorem 5.10 (Greenberg, Miller, Shen, and Westrick [68])**.** *If* $\dim(X) = 1$ *then there is an ML-random* Y *such that* $\{n : X(n) \neq Y(n)\}$ *has density* 0*.*

We remark that asymptotic density has seen a lot of work recently. We say that an algorithm Φ *coarsely* computes a set X if the density of $\{n \mid \Phi(n) \neq X(n)\}$ is zero. We say that Ψ *generically* computes X if $\Psi(n)\downarrow$ implies $\Psi(n) = X(n)$ and $\{n \mid \Psi(n) \uparrow\}$ has density 1. These concepts arose in combinatorial group theory. See e.g. Kapovich, Miasnikov, Schupp and Shpilrain [81].

Here is one example applied to a word problem. (The density here is natural as measured by the words generated by the generators.)

- Let $G = \langle a, b; R\rangle$ be any 2-generator group.
- Note Any countable group is embeddable in a 2-generator group so there are uncountably many such G.
- Let $F = \langle x, y \mid\rangle$ be the free group of rank 2.
- $H = G * \langle x, y\rangle := \langle a, b, x, y; R\rangle$ be the free product of G and F.
- Then the word problem for H is generically solvable in linear time.

To see this, take a long word w on the alphabet $\{a, b, x, y\}^{\pm 1}$, e.g. $abx^{-1}bxyaxbby$. Now erase the a, b symbols, freely reduce the remaining word on $\{x, y\}^{\pm 1}$, and if any letters remain, output "no". This partial algorithm gives no incorrect answers because if the image of w under the projection homomorphism to the free group F is not 1, then $w \neq 1$ in H.

$$abx^{-1}bxyaxbby \to x^{-1}xyxy \to yxy \neq 1$$

The successive letters on $\{x, y\}^{\pm 1}$ in a long random word $w \in H$ is a long random word in F which is not equal to the identity. So the algorithm answers "No" on a generic set and gives no answer if the image in F is equal to the identity. This method is called the *quotient method* and can be used for any $G = \langle X, R \rangle$ subgroup of K of finite index for which there is an epimorphism $K \to H$ hyperbolic and not virtually cyclic, to show generically solvable word problem.

There has been a lot of work understanding coarse and generic computability, especially in group theory, but also some in computability theory such as Jockusch and Schupp [79]. One very nice theorem from that paper.

Theorem 5.11 (Jockusch and Schupp [79]). *There exists a c.e. set A of density* 1 *which has no computable subset of density* 1*.*

As well, papers such as [73] and [5] have shown that coarse computability and algorithmic randomness are very closely related. Here is one typical theorem.

Theorem 5.12 (Hirschfeldt, Kuyper, Jockusch and Schupp [73]). *If A is ML-random and Bi is computable from every coarse description D of A, then B is K-trivial. Thus, if A is in 2-random*[18] *then B is computable.*

Work is ongoing. It would also be interesting to develop this kind of analysis in the setting of computable analysis. Here generic case and coarse complexity would likely be replaced by measure. In some sense this is discussed in the material on Ergodic Theory below.

The third question above is whether sources of randomness can be useful oracles. Here we are thinking in terms of complexity rather than just computability, so results such as Theorem 4.7 are not directly relevant. Allender and others have initiated a program to investigate the speedups that are possible when random sources are queried efficiently. Let R be the set of all random finite binary strings for either plain or prefix-free Kolmogorov complexity (e.g., $R = \{x : C(x) \geqslant |x|\}$). For a complexity class $\mathcal{C}$, let $\mathcal{C}^R$ denote the relativization of this class to R. So, for instance, for the class P of polynomial-time computable functions, P^R is the class of functions that can be computed in polynomial time with R as an oracle. (For references to the articles in this and the following theorem, see [1].)

Theorem 5.13 (Buhrman, Fortnow, Koucký, and Loff [30]; Allender, Buhrman, Koucký, van Melkebeek, and Ronneburger [3]; Allender, Buhrman, and Koucký [2]).

1. $\mathrm{PSPACE} \subseteq \mathrm{P}^R$.

2. $\mathrm{NEXP} \subseteq \mathrm{NP}^R$.

[18] They actually proved this for A being only weakly 2-random, meaning that A passes all ML-tests for which the modulus of convergence is not necessarily computable, only that $\lambda(U_n) \to 0$.

3. BPP $\subseteq \mathrm{P}_{tt}^{R}$ *(where the latter is the class of functions that are reducible to R in polynomial time via truth-table reductions, a more restrictive notion of reduction than Turing reduction).*

The choice of universal machine does have some effect on efficient computations, but we can quantify over all universal machines. In the result below, U ranges over universal prefix-free machines, and R_{K_U} is the set of random strings relative to Kolmogorov complexity defined using U.

Theorem 5.14 (Allender, Friedman, and Gasarch [4]; Cai, Downey, Epstein, Lempp, and Miller [31])**.**

1. $\bigcap_U \mathrm{P}_{tt}^{R_{K_U}} \subseteq \mathrm{PSPACE}$*.*

2. $\bigcap_U \mathrm{NP}^{R_{K_U}} \subseteq \mathrm{EXPSPACE}$*.*

We can also say that sufficiently random oracles will always accelerate *some* computations in the following sense. Say that X is *low for speed* if for any computable set A and any function t such that A can be computed in time $t(n)$ using X as an oracle, there is a polynomial p such that A can be computed (with no oracle) in time bounded by $p(t(n))$. That is, X does not significantly accelerate any computation of a computable set. Bayer and Slaman (see [20]) constructed noncomputable sets that are low for speed, but these cannot be very random.

Theorem 5.15 (Bienvenu and Downey [20])**.** *If X is Schnorr random, then it is not low for speed, and this fact is witnessed by an exponential-time computable set A.*

The proof of this theorem uses a kind of speed-up technique. Interestingly, whether generic sets speed up computations depends on $P \neq NP$?. (See [20], this result is due to Bayer in his PhD Thesis.)

5.4 Analysis and Ergodic Theory

The setting for this is the area of computable analysis. For simplicity, our spaces will have dense computable bases, like the rationals in $\mathbb{R}$. Classically we know that reals are Cauchy sequences, and in computable analysis, we regard a function f as being computable iff then f is *(Type 2) computable*[19] if there is a uniform algorithm Φ taking fast converging Cauchy sequences for input x (i.e. $q_k \in B(q_n, 2^{-n})$ for all $k > n$) to fast converging Cauchy sequences for output $f(x)$. Notice that

[19] Type 1 functions take $\mathbb{N}$ to itself, and type 2 take type 1 objects to themselves, so act on infinite sequences.

since the objects now are infinite, we won't have a finitely computable equality operator, rather we will have a computable distance function d in the sense that if x and y are reals, then uniformly in Cauchy sequences for each we can generate one for $d(x, y)$. Thus, computable analysis is an area that has developed tools for thinking about computability of objects like real-valued functions by taking advantage of separability. We can also relativize it, and it is then not difficult to see that a function is continuous iff it is computable relative to some oracle, basically because to define a continuous function we need only to specify its action on a countable collection of balls. Many results in this are show that our intuition about "good" vs "bad" is realized. For example we have the following (precise definitions are not important here.)

Theorem 5.16 (Pour-E and Richards see [117]). *In this setting an operator is computable iff it is bounded.*

A consequence of this is that there is a computable ODE with computable initial conditions and having no computable solution.

Mathematics is replete with results concerning almost everywhere behavior, and algorithmic randomness allows us to to turn such results into "quantitative" ones like the following.

Theorem 5.17 (Brattka, Miller, and Nies [27], also Demuth (1975, see [27]) for (2)).

1. *The reals at which every computable increasing function $\mathbb{R} \to \mathbb{R}$ is differentiable are exactly the computably random ones.*

2. *The reals at which every computable function $\mathbb{R} \to \mathbb{R}$ of bounded variation is differentiable are exactly the ML-random ones.*

Ergodic theory is another area that has been studied from this point of view. A measure-preserving transformation T on a probability space is *ergodic* if all measurable subsets E such that $T^{-1}(E) = E$ have measure 1 or 0. Notice that this is an "almost everywhere" definition. We can make this setting computable (and many systems arising from physics will be computable). One way to proceed is to work in Cantor space without loss of generality, since Hoyrup and Rojas [77] showed that any computable metric space with a computable probability measure is isomorphic to this space in an effective measure-theoretic sense. Then we can specify a computable transformation T as a computable limit of computable partial maps $T_n : 2^{<\omega} \to 2^{<\omega}$ with certain coherence conditions. We can also transfer definitions like that of ML-randomness to computable probability spaces other than Cantor space.

The following is an illustrative result. A classic theorem of Poincaré is that if T is measure-preserving, then for all E of positive measure and almost all x, we have $T^n(x) \in E$ for infinitely many n. For a class $\mathcal{C}$ of measurable subsets, x is a *Poincaré point* for T with respect to $\mathcal{C}$ if for every $E \in \mathcal{C}$ of positive measure, $T^n(x) \in E$ for infinitely many n. An *effectively closed* set is one whose complement can be specified as a computably enumerable union of basic open sets.

Theorem 5.18 (Bienvenu, Day, Mezhirov, and Shen [19]). *Let T be a computable ergodic transformation on a computable probability space. Every ML-random element of this space is a Poincaré point for the class of effectively closed sets.*

The reader might note that the hypothesis above did not say "T is measure-preserving." This case has been analysed, and Frankin and Towsner [59] proved that the Poincaré recurrence aligns itself to yet another randomness notion called "weak 2-randomness" which is defined exactly as we did for ML-randomness, except that we only asks that $\lambda(T_n) \to 0$, without knowing the modulus of convergence. Franklin and Towsner (and others) have analysed the Birkhoff recurrence theorem, and again various interpretations align to randomness notions. In terms of the physical interpretation of these results, Bravermann, Rojas and Schneider [28] have argued that, whilst noise makes short term behaviour difficult, in fact it allows prediction easier in the long term. While many of theorems of Ergodic theory have been analysed, including the Birkhoff, Poincaré, and von Neumann ergodic theorems, but some, like Furstenberg's ergodic theorem, are yet to be understood.

In analysis, there are other theorems which concern almost everywhere behaviour. One of the problems is how to address this effectively, since a function being computable relative to an oracle imples that it must be continuous. This means even step functions with a single computable step at, for instance, 0 is not computable relative to any oracle. There seem several ways around this one especially in the case of almost everywhere behaviour. One way was suggested by Hoyrup and Rojas [77] which defines a notion of *layerwise computability*, namely if $\{U_n \mid n \in \mathbb{N}\}$ is a ML-test then demanding that f is represented by a sequence of functions f_n such that f_n acts on (names of) reals X outside of U_n, and correctly gives an answer, such that if X passes the test, then $f(X) = f_n(X)$ for all n. This is also related to an earlier suggestion of Ko and Friedman [84] who suggested that it might be reasonable to look at f defined on a Kurtz test. The latter suggestion might certainly be easier in the setting of computational complexity. Again this is largely undeveloped.

5.5 The full circle: Turing, Borel and normality again

We return to Borel's notion of normality. This is a very weak form of randomness; polynomial-time randomness is more than enough to ensure absolute normality. Schnorr and Stimm [121] showed that a sequence is normal if and only if it satisfies a notion of randomness defined using with martingales defined by certain finite state automata[20]. Building examples of absolutely normal numbers is another matter, as Borel already noted. While it is conjectured that e, π, and all irrational algebraic numbers such as $\sqrt{2}$ are absolutely normal, *none* of these have been proved to be normal to *any* base. In his unpublished manuscript "A note on normal numbers", believed to have been written in 1938, Turing built a computable absolutely normal real, which is in a sense the closest we have come so far to obtaining an explicitly-described absolutely normal real. (His construction was not published until his Collected Works in 1992, and there was uncertainty as to its correctness until Becher, Figueira, and Picchi [15] reconstructed and completed it, correcting minor errors.[21])

An interesting aspect of Turing's construction is that he more or less anticipated Martin-Löf's work by looking at a collection of computable ML-style tests sensitive enough to make a number normal in all bases, yet insensitive enough to allow computable sequences to pass all such tests. We have seen that the strong law of large implies fixed blocks of digits should occur with the appropriate frequencies in a random sequence. Translating between bases results in correlations between blocks of digits in one base and blocks of digits in the other, which is why this extension allowed Turing to construct absolutely normal numbers. Turing made enough of classical measure theory computable to generate absolute normality, yet had the tests refined enough that computable sequence could still be "random".

This approach can also be thought of in terms of effective martingales, and its point of view has brought about a great deal of progress in our understanding of normality recently. For instance, Becher, Heiber, and Slaman [16] showed that absolutely normal numbers can be constructed in low-level polynomial time, and Lutz and Mayordomo (`arXiv:1611.05911`) constructed them in "nearly linear" time. Much of the work along these lines has been number-theoretic, connected

[20] An finite state automaton is a constrained Turing machine which will read an input tape and according to the symbol it is reading, and its internal state, transitions to a (perhaps) new state and moves on to the next symbol to the right of the input tape. The automaton accepts the input if when it gets to the last symbol it is in one of the designated accept states. It is possible to have a notion of randomness using "automatic" martingales using this idea and some definitions a wee bit too technical to include in this article.

[21] See `https://www-2.dc.uba.ar/staff/becher/publications.html` for references to the papers cited here and below.

to various notions of well-approximability of irrational reals, such as that of a *Liouville number*, which is an irrational α such that for every natural number $n > 1$, there are $p, q \in \mathbb{N}$ for which $|\alpha - \frac{p}{q}| < q^{-n}$. For example, Becher, Heiber, and Slaman [16] have constructed computable absolutely normal Liouville numbers. This work has also produced results in the classical theory of normal numbers, for instance by Becher, Bugeaud, and Slaman [14].

6 Summary

We have given a reasonably self-contained, if perhaps idiosyncratic, account of many of the basics of algorithmic randomness as well as a number of recent applications. Clearly the area is in a state of rapid development and we have necessarily left a lot out, but we suggest that the reader follows up the references for more details. I have definitely left a huge amount out, such as randomness lower down, in complexity classes, and also higher up, with Π^1_1-randomness, a concept going back to Martin-Löf but having a lot of interest recently, in papers such as Greenberg and Monin [69] and Hjorth-Nies [74]. Nor do I discuss the developing area of ML-randomness and quantum physics such as [111]; nor Brownian motion such as [57, 116]. These areas are all in rapid growth. The present article should at least give pointers to the aspirations and scope of such studies.

References

[1] Eric Allender *The complexity of complexity*, In Computability and Complexity, volume 10010 of Lecture Notes in Comput. Sci., pp 79–94. Springer, Cham, 2017.

[2] Eric Allender, Harrry Buhrman, and Michael Koucký *What can be efficiently reduced to the Kolmogorov-random strings?* Annals of Pure and Applied Logic, 138(1-3) 2006, pp 2–19.

[3] Eric Allender, Harry Buhrman, Michael Koucký, Dieter van Melkebeek, and Detlef Ronneburger *Power from random strings,* SIAM Journal on Computing, 35(6) 2006, pp 1467–1493.

[4] Eric Allender, Luke Friedman, and William Gasarch *Limits on the computational power of random strings,* Information and Computation, 222 2013, pp 80–92.

[5] Uri Andrews, Mingzhong Cai, David Diamondstone, Carl Jockusch, and Steffen Lemp *Asymptotic density, computable traceability and 1-randomness* Fundamenta Mathematicae, 234 2016, pp 41–53.

[6] Luis Antunes, Lance Fortnow, Dieter van Melkelbeck, and Vijay Vinochandram *Computational depth: Concept and applications*, Theoretical Computer Science, 354 2006, pp 391–404.

[7] Luis Antunes, Armando Matos, Andre Souto, and Paul Vitanyi *Depth as randomness deficiency*, Theory of Computing Systems, 45 2009, pp 724–739.

[8] Krishna Athreya, John Hitchcock, Jack Lutz, and Elvira Mayordomo *Effective strong dimension in algorithmic information and computational complexity*, SIAM Journal on Computing, 37(3) 2007, pp 671–705.

[9] George Barmpalias and Andrew Lewis-Pye *Monotonous betting strategies in warped casinos*, Information and Computation, 271, 2020, to appear.

[10] George Barmpalias and Andrew Lewis-Pye *Differences of halting probabilities*, Journal of Computer and System Sciences, 89 2017, pp 349–360.

[11] George Barmpalias and Andrew Lewis-Pye *Optimal redundancy in computations from random oracles*, Journal of Computer and System Sciences, 92 2018, pp 1–8.

[12] George Barmpalias and Rod Downey *Kobayashi compressibility*, Theoretical Computer Science A, 675 2017, pp 89–100.

[13] Bruno Bauwens *Relating and contrasting plain and prefix Kolmogorov complexity*, Theory of Computing Systems, 58 2015, pp 482–501.

[14] Veronica Becher, Yves Bugeaud, and Theodore Slaman *On simply normal numbers to different bases*, Mathematische Annalen, 364(1–2) 2016 pp 125–150.

[15] Veronica Becher, Santiago Figueira, and Rafael Picchi *Turing's unpublished algorithm for normal numbers*, Theoretical Computer Science, 377(1–3) 2007, pp 126–138.

[16] Veronica Becher, Pablo Heiber, and Theodore Slaman *A polynomial-time algorithm for computing absolutely normal numbers*, Information and Computation, 232 2013, pp 1–9.

[17] Charles Bennett, Peter Gács, Ming Li, Paul Vitanyi, and Wojciech Zurek *Information distance*, arxiv.org/abs/1006.3520v1.

[18] Charles Bennett *Logical depth and physical complexity*, In Rola Herken, editor, The Universal Turing Machine: a Half-Century Survey, pp 227–257, Oxford U. Press, Oxford, 1992.

[19] Laurent Bienvenu, Adam Day, Ilya Mezhirov, and Alexander Shen *Ergodic-type characterizations of algorithmic randomness*, In Programs, Proofs, Processes, volume 6158 of Lecture Notes in Comput. Sci., pages 49–58. Springer, Berlin, 2010.

[20] Laurent Bienvenu and Rod Downey *On Low for Speed Oracles*, In R. Niedermeier and B. Vallée, editors, 35th Symposium on Theoretical Aspects of Computer Science (STACS 2018), volume 96 of Leibniz International Proceedings in Informatics (LIPIcs), pages 15:1–15:13, Germany, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018.

[21] Laurent Bienvenu, An. A. Muchnik, Alexander Shen, and Nicolai Vereshchagin *Limit complexities revisited*, In 25th International Symposium on Theoretical Aspects of Computer Science. STACS 2008, pp 73–84 . Leibniz International Proceedings in Informatics, 2008.

[22] Laurent Bienvenu and Ludovic Patey *Diagonally non-computable functions and fireworks*, Information and computation, 253, 2017, pp 64–77.

[23] Laurent Bienvenu, Andrei Romashchenko, Alexander Shen, Antoine Taveneaux, Stijn Vermeeren *The axiomatic power of Kolmogorov complexity*, Annals of Pure and Applied Logic, 165(9), 2014, pp 1380–1402.

[24] Laurent Bienvenu, Frank Stephan, and Jason Teutsch *How powerful are integer valued martingales?*, In Proceedings CIE 2010, 2012.

[25] Bella Bollobas Random graphs, Cambridge Univ. Press, Cambridge, 2001.

[26] Emil Borel *Les probabilités dénombrables et leurs applications arithmétiques*, Rendiconti del Circolo Matematico di Palermo (1884-1940), 27 1909, pp 247–271.

[27] Vasco Brattka, Joseph Miller, and Andre Nies *Randomness and differentiability*, Transactions of the American Mathematical Society 368, 2016, pp 581–605.

[28] Mark Braverman, Cristobal Rojas, and Jonathan Schnieder *Space-bounded church-turing thesis and computational tractability of closed systems*, Phys. Rev. Lett. 115. Aug 28;115(9):098701. Epub 2015 Aug 27, 2015.

[29] Mark Braverman and Michael Yampolsky *Computability of Julia sets*, Moscow Mathematical Journal, 8(2) 2008, pp 185–231.

[30] Harry Buhrman, Michal Koucký, Lance Fortnow, and Bruno Loff *Derandomizing from random strings*, In 25th Annual IEEE Conference on Computational Complexity—CCC 2010, pages 58–63. IEEE Computer Soc., Los Alamitos, CA, 2010.

[31] Mingzhong Cai, Rod Downey, Rachel Epstein, Steffen Lempp, and Joseph Miller *Random strings and truth-table degrees of Turing complete c.e. sets*, Logical Methods in Computer Science, 10(3) 2014, pp 3-15.

[32] Cristian Calude, Peter Hertling, Bakhadyr Khoussainov, and Yongge Wang *Recursively enumerable reals and Chaitin Ω numbers*, In M. Morvan, C. Meinel, and D. Krob, editors, STACS 98. 15th Annual Symposium on Theoretical Aspects of Computer Science. Paris, France, February 25–27, 1998. Proceedings, volume 1373 of Lecture Notes in Computer Science, pages 596–606. Springer, Berlin, 1998.

[33] Gregory Chaitin *Computational complexity and Gödel's incompleteness theorem*, ACM SIGACT News, 9:11–12, 1971.

[34] Gregory Chaitin *A theory of program size formally identical to information theory*, Journal of the Association for Computing Machinery, 22, 1975, pp 329–340.

[35] Gregory Chaitin *Information-theoretical characterizations of recursive infinite strings*, Theoretical Computer Science, 2 1976, pp 45–48.

[36] Gregory Chaitin *Nonrecursive infinite strings with simple initial segments*, IBM Journal of Research and Development, 21 1977 pp 350–359.

[37] Peter Colak, Rod Downey, and Noam Greenberg *Strong jump traceablilty I, the computably enumerable case*, Advances in Mathematics, 217, 2008, pp 2045–2074.

[38] Alonzo Church *On the concept of a random sequence*, Bulletin of the American Mathematical Society, 46, 1940, pp 130–135.

[39] Chris Conidis *A real of strictly positive effective packing dimension that does not compute a real of effective packing dimension one*, The Journal of Symbolic Logic, 77(2) 2012 pp 447–474.

[40] Adam Day *Process and truth-table characterizations of randomness*, Theoretical

Computer Science, 452, 2012, pp 47–55.

[41] Karel de Leeuw, Edward Moore, Claude Shannon, and Norman Shapiro Computability by probabilistic machines. Automata Studies, volume 34 of Annals of Mathematics Studies. Princeton University Press, 1956.

[42] Oswald Demuth *The differentiability of constructive functions of weakly bounded variation on pseudo numbers*, Commentationes Mathematicae Universitatis Carolinae, 16(3), 1975, pp 583–599.

[43] Rod Downey and Noam Greenberg *Turing degrees of reals of positive effective packing dimension*, Information Processing Letters, 108, 2008, pp 298–303.

[44] Rod Downey and Noam Greenberg *Pseudo-jump inversion, upper cone avoidance, and strong jump-traceability*, Advances in Mathematics, 237 2013, pp 252–285.

[45] Rod Downey, Noam Greenberg and Andrew Tangarra *Divergence and Convergence of Random Series*, in preparation.

[46] Rod Downey and Evan Griffiths *Schnorr randomness*, The Journal of Symbolic Logic, 69, 2004, pp 533–554.

[47] Rod Downey and Denis Hirschfeldt Algorithmic Randomness and Complexity, Theory and Applications of Computability. Springer, New York, 2010.

[48] Rod Downey and Denis Hirschfeldt *Algorithmic randomness*, Communications of The Association for Computing Machinery, 62, 2019, pp 70–80.

[49] Rod Downey and Denis Hirschfeldt *Computability and randomness*, Notices of the American Mathematical Society 66 (7) 2019, pp 1001-1012.

[50] Rod Downey, Denis Hirschfeldt, Joseph Miller, and Andre Nies *Relativizing Chaitin's halting probability*, Journal of Mathematical Logic, 5, 2005, pp 167–192.

[51] Rod Downey, Denis Hirschfeldt, and Andre Nies *Randomness, computability and density*, SIAM J. Comput. 31, 2002, pp 1169–1183.

[52] Rod Downey, Denis Hirschfeldt, Andre Nies, and Frank Stephan *Trivial reals*, In Rod Downey, Decheng Ding, Shi Tung, Yu Qiu, and Mariko Yasugi, editors, Proceedings of the 7th and 8th Asian Logic Conferences. Held in Hsi-Tou, June 6–10, 1999 and Chongqing, August 29-September 2, 2002, pages 103–131. Singapore University Press and World Scientific, Singapore, 2003.

[53] Bruno Durand, Leonid Levin, and Alexander Shen *Complex tilings*, The Journal of Symbolic Logic, 73(2), 2006, pp 593–613.

[54] Bruno Durand, Andrei Romashchenko, and Alexander Shen *Fixed-point tile sets and their applications*, Journal of Computer and System Sciences, 78(3), 2012 pp 731–764.

[55] Santiago Figueira, Frank Stephan, and Andre Nies *Lowness properties and approximations of the jump*, Ann. Pure Applied Logic, 152, 2008, pp 51–66.

[56] Lance Fortnow, John Hitchcock, A. Pavan, N. Vinochandran, and Fengming Wang. *Extracting Kolmogorov complexity with applications to dimension zero-one laws*, In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, Automata, Languages and Programming. 33rd International Colloquium, ICALP 2006. Venice, Italy, July 10–14, 2006, Proceedings, Part I, volume 4051 of Lecture Notes in Computer Science,

pages 335–345. Springer, Berlin, 2006.

[57] Willem Fouche *Diophantine properties of Brownian motion: recursive aspects*, In Viasco Brattka, Hannes Diener, and Dieter Spreen, editors, Logic, Computation, Hierarchies, pages 139–156. Springer-Verlag, 2014.

[58] Johanna Franklin and Keng Meng Ng *Difference randomness*, Proceedings of the American Mathematical Society, 139(1), 2011, pp 345–360.

[59] Johanna Franklin and Henry Towsner *Randomness and non-ergodic systems*, Moscow Mathematical Journal, 14(4), 2014, pp 711–744.

[60] Peter Gács *On the symmetry of algorithmic information*, Soviet Mathematics Doklady, 15 1974, pp 1477âĂŞ1480.

[61] Peter Gács *Every set is reducible to a random one*, Information and Control, 70, 1986, pp 186–192.

[62] Michael Garey and David Johnson. Computers and Intractability, Freeman, 1979.

[63] Noam Greenberg, Denis Hirschfeldt, and Andre Nies *Characterizing the strongly jump-traceable sets via randomness*, Advances in Mathematics, 231, 2012, pp 2252âĂŞ2293.

[64] Noam Greenberg, Joseph Miller, Benoit Monin, and Daniel Turetsky *Two more characterizations of K-triviality*, Notre Dame Journal of Formal Logic, 59(2), 2018, pp 189âĂŞ195.

[65] Noam Greenberg, Joseph Miller, Andre Nies, and Daniel Turetsky *Martin-Löf reducibility and cost functions*, submitted.

[66] Noam Greenberg and Joseph Miller *Lowness for Kurtz randomness*, The Journal of Symbolic Logic, 74, 2009, pp 665âĂŞ678.

[67] Noam Greenberg and Joseph Miller *Diagonally non-recursive functions and effective Hausdorff dimension*, Bulletin of the London Mathematical Society, 43(4), 2011, pp 636–654.

[68] Noam Greenberg, Joseph Miller, Alexande Shen, and Linda Brown Westrick *Dimension 1 sequences are close to randoms*, Theoretical Computer Science, 705, 2018, pp 99–112.

[69] Noam Greenberg and Benoit Monin *Higher randomness and genericity*, Forum of Mathematics: Sigma 5, 41 pages, 2017.

[70] Noam Greenberg and Andre Nies *Benign cost functions and lowness notions*, Journal of Symbolic Logic 76, 2011, pp 289–312.

[71] Noam Greenberg and Daniel Turetsky *Strong jump-traceability*, Bulletin of Symbolic Logic, 24, 2018, pp 147–164.

[72] Daniel Hammer, Andrei Romashchenko, Alexander Shen, and Nikolai Vereshchagin *Inequalities for Shannon entropies and Kolmogorov complexities*, In IEEE Conference on Computational Complexity, 1997. Journal Version Journal of Computing and System Sciences 60, 2000, pp 442-464.

[73] Denis Hirschfeldt, Carl Jockusch, Rutger Kuyper, and Paul Schupp *Coarse reducibility and algorithmic randomness*, J. Symbolic Logic, 81, 2016, pp 1028 – 1046.

[74] Gregory Hjorth and Andre Nies *Randomness via effective descriptive set theory*, J.

London Math Soc 75, 2007, pp 495–508.

[75] Michael Hochman and Tom Meyerovitch. *A characterization of the entropies of multidimensional shifts of finite type*, Annals of Mathematics. Second Series, 171(3), 2010, pp 2011–2038.

[76] Rupert Hölzl, Wolfgang Merkle, Frank Stephan, and Liang Yu *Chaitin's ω as a continuous function*, Journal of Symbolic Logic, 85, 2020, pp 486-510.

[77] Mathieu Hoyrup and Cristobal Rojas *Computability of probability measures and Martin-Löf randomness over metric spaces*, Information and Computation, 207(7), 2009, pp 830–847.

[78] Russel Impagliazzo and Avi Wigderson P = BPP *if* E *requires exponential circuits: derandomizing the XOR lemma*, In STOC '97 (El Paso, TX), pages 220–229. ACM, New York, 1999.

[79] Carl Jockusch and Paul Schupp *Generic computability, Turing degrees, and asymptotic density*, Journal of the London Mathematical Society, 85 (2), 2012, pp 472–490.

[80] Jean-Pierre Kahane Some random series of functions, Cambridge University Press, 2003.

[81] Ilya Kapovich, Alexei Myasnikov, Paul Schupp, Vladimir Shpilrain *Generic case complexity, decision problems in group theory and random walks*, J. Algebra, 264, 2003, pp 665âĂŞ694.

[82] Bjorn Kjos-Hanssen, Wolfgang Merkle, and Frank Stephan *Kolmogorov complexity and the recursion theorem*, In B. Durand and W. Thomas, editors, STACS 2006. Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science, Marseille, France, February 23–25, 2006, volume 3884 of Lecture Notes in Computer Science, pages 149–161. Springer-Verlag, Berlin, 2006.

[83] Bjorn Kjos-Hanssen, Andre Nies, and Frank Stephan *Lowness for the class of Schnorr random sets*, SIAM Journal on Computing, 35, 2005, pp 647–657.

[84] Ker-I. Ko and Harvey Feideman *Computational complexity of real functions*, Theoretical Computer Science, 1982, pp 323–352.

[85] Kojiro Kobayashi. *On compressibility of infinite sequences*, Technical Report C-34, Depart- ment of information sciences, Tokyo Institute of Technology. Series C. 1993.

[86] Shira Kritchman and Ran Raz *The surprise examination paradox and the second incompleteness theorem*, Notices of the American Mathematical Society, 57(11), 2010, pp 1454–1458.

[87] Antonin Kučera *Measure, Π^0_1 classes, and complete extensions of PA*, In H.-D. Ebbinghaus, G. H. Müller, and G. E. Sacks, editors, Recursion Theory Week. Proceedings of the Conference Held at the Mathematisches Forschungsinstitut in Oberwolfach, April 15–21, 1984, volume 1141 of Lecture Notes in Mathematics, pages 245–259. Springer, Berlin, 1985.

[88] Antonin Kučera and Theodore Slaman *Randomness and recursive enumerability*, SIAM Journal on Computing, 31, 2001, pp 199–211.

[89] Stuart Kurtz Randomness and Genericity in the Degrees of Unsolvability. Ph.D. dissertation, University of Illinois at Urbana–Champaign, 1981.

[90] Ming Li and Paul Vitányi An Introduction to Kolmogorov Complexity and its Applications. Texts in Computer Science. Springer, New York, third edition, 2008.

[91] Jack Lutz *Gales and the constructive dimension of individual sequences*, In U. Montanari, J. D. P. Rolim, and E. Welzl, editors, Automata, Languages and Programming. 27th International Colloquium, ICALP 2000. Geneva, Switzerland, July 9–15, 2000. Proceedings, volume 1853 of Lecture Notes in Computer Science, pages 902–913. Springer, Berlin, 2000.

[92] Jack Lutz *The dimensions of individual strings and sequences*, Information and Computation, 187, 2003, pp 49–79.

[93] Jack Lutz and Neil Lutz *Algorithmic information, plane Kakeya sets, and conditional dimension*, ACM Transactions on Computation Theory, 10(2):Art. 7, 22, 2018.

[94] Neil Lutz. *Fractal intersections and products via algorithmic dimension*, In 42nd International Symposium on Mathematical Foundations of Computer Science, volume 83 of LIPIcs. Leibniz Int. Proc. Inform., pages Art. No. 58, 12. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2017.

[95] Neil Lutz and Don Stull *Bounding the dimension of points on a line*, In Theory and Applications of Models of Computation, volume 10185 of Lecture Notes in Comput. Sci., pages 425–439. Springer, Cham, 2017.

[96] Per Martin-Löf *The definition of random sequences*, Information and Control, 9, 1966 pp 602–619.

[97] Elvira Mayordomo *A Kolmogorov complexity characterization of constructive Hausdorff dimension*, Information Processing Letters, 84, 2002 pp 1–3.

[98] Wolfgang Merkle, Nenad Mihailovíc, and Theodore Slaman *Some results on effective randomness*, Theory of Computing Systems, 39, 2006, pp 707–721.

[99] Jochen Messner and Thomas Thierauf. *A Kolmogorov complexity proof of the Lovász local lemma for satisfiability*, Theoretical Computer Science, 461, 2012, pp 55–64.

[100] Joseph Miller *Kolmogorov random reals are 2-random*, Journal of Symbolic Logic, 69, 2004, pp 907–913.

[101] Joseph Miller *The K-degrees, low for K-degrees, and weakly low for K sets*, Notre Dame Journal of Formal Logic, 50, 2010, pp 381–391.

[102] Joseph Miller *Extracting information is hard: a Turing degree of non-integral effective Hausdorff dimension*, Advances in Mathematics, 226(1), 2011, pp 373–384.

[103] Joseph Miller *On work of Barmpalias and Lewis-Pye: A derivation on the d.c.e. reals*, In Computability and Complexity: Essays Dedicated to Rodney G. Downey on the Occasion of His 60th Birthday, volume 10010 of Lecture Notes in Computer Science, page 644âĂŞ659. Springer-Verlag, 2016.

[104] Joseph Miller and Liang Yu *On initial segment complexity and degrees of randomness*, Transactions of the American Mathematical Society, 360, 2008, pp 3193–3210.

[105] Philippe Moser *On the polynomial septh of various sets of random strings*, Theoretical Computer Science, 477, 2013, pp 96–108.

[106] Philippe Moser and Frank Stephan *Depth, highness and dnr degrees*, In Fundamentals of Computation Theory, Twentieth International Symposium, FCT 2015, Gdansk,

Poland, August 17âĂŞ19, 2015, Proceedings, Springer LNCS 9210, pp 81-94, 2015.

[107] Andrei A. Muchnik, Alexi Semenov, and Vladimir Uspensky *Mathematical metaphysics of randomness*, Theoretical Computer Science, 207, 1998, pp 263–317

[108] Andre Nies *Lowness properties and randomness*, Advances in Mathematics, 197, 2005, pp 274–305.

[109] Andre Nies Computability and Randomness, volume 51 of Oxford Logic Guides. Oxford University Press, Oxford, 2009.

[110] Andre Nies *Calculus of cost functions*, In Mariya Soskova and Barry Cooper, editors, The Incomputable, pages 183–216, 2017.

[111] Andre Nies and Volkher Scholz *Martin-Löf random quantum states*, Submitted, arXiv: 1709.08422.

[112] Andre Nies, Frank Stephan, and Sebastiaan Terwijn *Randomness, relativization, and Turing degrees*, The Journal of Symbolic Logic, 70, 2005, pp 515–535.

[113] Raymond Paley and Antoni Zugmund *On some series of functions (1)*, Mathematical Proceedings of the Cambridge Philosophical Society, 26, 1930, pp 337-357.

[114] Raymond Paley and Antoni Zugmund *On some series of functions (2)*, Mathematical Proceedings of the Cambridge Philosophical Society 26, 1930, pp 459–474.

[115] Raymond Paley and Antoni Zugmund *On some series of functions, (3)*, Mathematical Proceedings of the Cambridge Philosophical Society 28, 1932, pp 190–205.

[116] Paul Potgieter *Algorithmically random series and Brownian motion*, Annals of Pure and Applied Logic, 169, 2018, pp 1210âĂŞ1226.

[117] Marion Pour-El and Ian Richards Computability in Analysis and Physics. Springer-Verlag, 1989.

[118] Hartley Rogers Jr. The Theory of Recursive Functions and Effective Computability, MIT Press, 1967.

[119] Gerald Sacks Degrees of Unsolvability. Annals of Mathematics, Studies. Princeton University Press, 1963.

[120] Claus Schnorr Zufälligkeit und Wahrscheinlichkeit. Eine algorithmische Begründung der Wahrscheinlichkeitstheorie, volume 218 of Lecture Notes in Mathematics. Springer-Verlag, Berlin–New York, 1971.

[121] Claus Schnorr and H. Stimm *Endliche Automaten und Zufallsfolgen*, Acta Informatica, 1(4), 1971/2, pp 345–359.

[122] Alexander Shen, Vladimir Uspensky and Nicolai Vereshchagin Kolmogorov Complexity and Algorithmic Randomness, American Mathematical Society, 2017.

[123] Stephen Simpson *Symbolic dynamics: entropy = dimension = complexity*, Theory of Computing Systems, 56(3), 2015 pp 527–543.

[124] Robert Soare Recursively Enumerable Sets and Degrees, Springer-Verlag, 1987.

[125] Robert Solovay. Draft of paper (or series of papers) on Chaitin's work, Unpublished notes, May 1975. 215 pages.

[126] Frank Stephan *Martin-Löf random sets and PA-complete sets*, In Z. Chatzidakis, P. Koepke, and W. Pohlers, editors, Logic Colloquium '02, volume 27 of Lecture

Notes in Logic, pp 342–348, Association for Symbolic Logic and A K Peters, Ltd., La Jolla, CA and Wellesley, MA, 2006.

[127] Sebastiaan Terwijn and Domenico Zambella *Algorithmic randomness and lowness*, The Journal of Symbolic Logic, 66, 2001, pp 1199–1205.

[128] Alan Turing *On computable numbers with an application to the Entscheidungsproblem*, Proceedings of the London Mathematical Society, 42, 1936, pp 230–265. Correction in Proceedings of the London Mathematical Society 43, 1937, pp 544–546.

[129] Nicolai Vereshchagin *Kolmogorov complexity conditional to large integers*, Theoretical Computer Science, 271, 2002, pp 59–67.

[130] Jean Ville Etude Critique de la Notion de Collectif. Monographies des Probabilités. Calcul des Probabilités et ses Applications. Gauthier-Villars, 1939.

[131] Richard von Mises *Grundlagen der Wahrscheinlichkeitsrechnung*, Mathematische Zeitschrift, 5, 1919, pp 52–99.

[132] Abraham Wald *Sur la notion de collectif dans la calcul des probabilités*, Comptes Rendus des Seances de l'Académie des Sciences, 202, 1936, pp 180–183.

[133] Abraham Wald *Die Wiederspruchsfreiheit des Kollektivbegriffes der Wahrscheinlichkeitsrechnung*, Ergebnisse eines Mathematischen Kolloquiums, 8, 1937 pp 38–72.

[134] Liang Yu, Decheng Ding, and Rod Downey *The Kolmogorov complexity of random reals*, Annals of Pure and Applied Logic, 129, 2004, pp 163–180.

[135] Domenico Zambella *On sequences with simple initial segments*, In Technical Report, The Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 1990.

[136] Marius Zimand *Two sources are better than one for increasing the Kolmogorov complexity of infinite sequences*, Theory of Computing Systems, 46, 2010, pp 707–722.