DOI:10.1145/3319408

Tracing some of the latest advancements in algorithmic randomness.

BY ROD DOWNEY AND DENIS R. HIRSCHFELDT

Algorithmic Randomness

CLASSICAL PROBABILITY THEORY gives all sequences of fair coin tosses of the same length the same probability. On the other hand, when considering sequences such as

10101010101010101010101010101010101010...

and

10110101110101011110000101010001011 1...,

none but the most contrarian among us would deny that the second (obtained by the first author by tossing a coin) is more random than the first. Indeed, we might well want to say that the second sequence is entirely random, whereas the first one is entirely nonrandom. But what are we to make in this context of, say, the sequence obtained by taking our first sequence, tossing a coin for each bit, and if the coin comes up heads, replacing that bit by the corresponding one in the second sequence? There are deep and fundamental questions involved in trying to understand why some sequences should count as "random," or "partially random," and others as "predictable," and how we can transform our intuitions about these concepts into meaningful mathematical notions.

One goal of the theory of algorithmic randomness is to give meaning to the notion of a random *individual* (infinite) sequence. Questions immediately arise: How should we define randomness? How can we measure whether one sequence is more random than another? How are computational power and randomness related? Is a theory of randomness for individual sequences at all useful? How does such a theory relate to classical probability theory?

The modern development of the theory of algorithmic randomness goes back to the 1960s (with even earlier roots, as we will discuss), but there has been a particular surge of development in the last couple of decades. In this article, we hope to give some of the flavor of this work, though we will be able to mention only a few samples of what is by now a vast area of research. Our book,[11] for example, is over 800 pages long and still manages to cover only a fraction of the area ... Another book covering some of the work we discuss here is Nies.[35] Li and Vitányi[27] is broader in scope and focuses more on Kolmogorov complexity of finite strings.

For simplicity, we assume sequences are binary unless we say otherwise. We use the terms *sequence* for an infinite sequence and *string* for a finite one. We write $|\sigma|$ for the length of a string σ,

» key insights

- **Computational theory can be used to give precise meaning to the notion of randomness for individual objects such as finite strings, infinite sequences, and real numbers.**
- **This theory also allows us to calibrate partial randomness. Notions of partial randomness can then be used to assign dimensions to individual points, as opposed to sets of points, leading to useful tools for establishing new results in areas such as the theory of fractal dimensions.**
- **It also gives us insight into how much and what kinds of randomness are needed for certain results in computer science and mathematics.**

IMAGE BY DUKENOD

write $X(n)$ for the nth bit of the sequence X (beginning with the 0th bit $X(0)$), and write $X\upharpoonright n$ for the string consisting of the first n bits of X. We identify a real number x in the interval [0, 1] with the binary sequence X such that $x = 0.X$. There are reals that have two binary expansions, but they are all rational and will not be relevant.

Historical Roots

Borel. In the beginning of the 20th century, Émile Borel was interested in sequences that satisfy the *(strong) law of large numbers*, which says that if we repeat an experiment with a numerical result many times, the average value of the result should be close to its expected value. If we toss a fair coin many times, for example, we expect the frequency of heads to be about $\frac{1}{2}$. Let X be a sequence representing infinitely many such tosses. After s many coin tosses, we can see how we are doing so far by looking at how many heads we have seen in the first s tosses compared to s, that is, the ratio

$$\frac{|\{X(k)=1 \mid k<s\}|}{s},$$

where we think of a 1 as representing heads. If this is indeed a fair coin, this ratio should get closer and closer to $\frac{1}{2}$ as s increases. Moreover for the *strong* law, any length k subsequence, such as 1011 (of length 4), should appear with frequency approaching $\frac{1}{2^k}$.

More generally, we say that an n-ary sequence sequence X is *(Borel) normal* if it has the same property relative to an "n-sided coin," in other words, if for any length m sequence $\sigma = a_1 a_2 \ldots a_m$ of digits between 0 and $n-1$,

$$\lim_{s\to\infty} \frac{\text{number of times } \sigma \text{ appears as a substring of } X\upharpoonright s}{s} = \frac{1}{n^m}.$$

Borel defined a real number to be *normal to base n* if its base n representation is normal, and *absolutely normal* if it is normal to *every* base. Borel observed that almost every real number is absolutely normal. Mathematically, this fact can be expressed by saying the collection of absolutely normal numbers has Lebesgue measure 1, which corresponds to saying that if we threw a dart at the real line, with probability 1, it would hit an absolutely normal number. We would thus expect a random sequence to be normal, and indeed (recalling that we identify the sequence X with the real number $0.X$) we would expect a random sequence to be absolutely normal.

Von Mises and Ville. The late 1920s and early 1930s saw the development, particularly by Andrey Kolmogorov, of an adequate foundation for probability theory, using measure theory and based on the idea of the expected behavior of events in a probability space. This theory does not give any meaning to the idea of randomness of an individual object, such as a particular sequence of coin tosses. Tossing a fair coin n times takes place in a "space of possibilities" (in this case, the collection of all binary strings of length n), and we assign any sequence of length n the probability 2^{-n} of occurring. For example, as we are taught in school, any particular sequence of three coin tosses occurs with probability $2^{-3} = \frac{1}{8}$.

In the infinite case, we might look at the event that a sequence has a certain string, say 101, as an initial segment. The probability that we begin a sequence of coin tosses with heads, tails, heads is $2^{-3} = \frac{1}{8}$. The mathematical way to express this fact is that the (*uniform*) *measure* (also known as the *Lebesgue measure*) of the set of sequences beginning with 101 is 2^{-3}, or, more generally, the measure of the set of sequences beginning with any particular string of length n is 2^{-n}. Probability theory is of course a vast and complex field, but for our purposes, this simple example suffices.

It is less commonly known that Kolmogorov's work came after earlier attempts to give meaning to the notion of randomness for individual objects such as infinite sequences. This idea is completely contrary to the approach in which all sequences are equally likely, but is quite reasonable when thinking about the difference between sequences like the two that open this article. The question is how to differentiate between a sequence like 01101 1100101110111100010011010101111100..., the base 2 version of *Champernowne's sequence*, obtained by listing the binary representations of the natural numbers in order and clearly nonrandom, and one arising from a random source. There are tests we can apply to a sequence to try to verify its apparent randomness. For example, a random sequence should be normal in the sense of the previous section. However, that is not a sufficient condition, as the aforementioned sequence is known to be normal to base 2, but is highly predictable.

In 1919, Richard von Mises[a] attempted to give a definition of randomness for a sequence X based upon a generalization of the law of large numbers. His idea was to require normality not only of X itself, but also of (certain) infinite subsequences of X. The point here is that the base 2 Champernowne sequence is normal, but if we computably select every $[g(n) = 1 + \sum_{j\le n} j(2^j - 2^{j-1})]$-th bit of this sequence, the resulting subsequence 1111 ... is no longer normal. It is not reasonable that selecting such bits of a random sequence should result in all 1s, so our sequence fails this randomness test.

Von Mises generalized this idea as follows. Let $f : \mathbb{N} \to \mathbb{N}$ be an increasing function. We think of f as a *selection function* for determining a subsequence of a given sequence. That is, $f(i)$ is the i^{th} place selected in forming this subsequence. In the law of large numbers itself, where we consider the entire sequence, $f(\mathrm{i}) = i$. In the nonrandomness argument in the previous paragraph, $f(i) = g(i)$. Von Mises proposed replacing the ratio $\frac{|\{X(k)=1 \mid k<s\}|}{s}$ coming from the law large numbers by

$$\frac{|\{X(f(k))=1 \mid k<s\}|}{s},$$

the ratio of the number of *selected places* at which X has value 1 to the total number of selected places. For base 2 and each choice of f, the requirement that this ratio approach $\frac{1}{2}$ as s goes to infinity constitutes a randomness test.

So when should X be regarded as random? We could perhaps try to say that X is random if and only if it passes this test for all possible selection functions, reflecting the idea that in a sequence of coin tosses, there should be no way to select a subsequence ahead of time that will have a greater proportion of heads than tails. There is a big problem with this idea, though. No sequence X can be random for *all* selection functions. As

a See Downey and Hirschfeldt[11] for references to this and other sources mentioned in this section.

any nontrivial X has infinitely many 0s, there is an f that chooses the positions of the 0's of X in increasing order. But surely this counterexample is unfair to the spirit of von Mises' idea: we are trying to capture the notion that we should not be able to *predict* the values of bits of X, and this f is chosen *after* defining X. It is always easy to predict the answer if you know it in advance! The question then is what kinds of selection functions should be allowed, to capture the notion of prediction. A reasonable intuition is that prediction is somehow a computational process, and hence from a modern perspective we might want to restrict ourselves to *computable* selection functions, a suggestion later made by Alonzo Church.

Von Mises' work predated the definition of computable function, however, so he had no canonical choice of "acceptable selection rules" and left his definition mathematically vague. But Abraham Wald showed that for any countably infinite collection of selection functions, there is a sequence that is random in the sense of passing all tests corresponding to the functions in this collection.

However, von Mises' program was dealt a major blow in 1939 by Jean Ville, who showed that for any countable collection of selection functions, there is a sequence X that passes all of the resulting tests, but such that for each n, there are always more 0s than 1s in $X\restriction n$. If we were told that there would always be more tails than heads in a sequence of coin flips, we would not believe the coin to be a standard one, and could use this information to make some money betting on its flips. Thus, Ville's sequence is random in the sense of von Mises, but certainly not random in the intuitive sense.

Ville suggested adding versions of another law (the law of the iterated logarithm) to the list of tests that a sequence would need to pass to be considered random. Perhaps von Mises' tests together with these additional tests would capture the notion of algorithmic randomness. But this all begins to look very ad hoc, and immediately raises the natural question of whether there is a Ville-like counterexample for this new set of laws. (As it turns out, there is, as discussed, for example, in Downey and Hirschfeldt.[11])

We are abandoning the idea of absolute randomness in some metaphysical sense in favor of a notion of algorithmic randomness, where we use tools from computability theory to define and quantify randomness.

Notice that in these discussions, we are abandoning the idea of *absolute randomness* in some metaphysical sense in favor of a notion of *algorithmic randomness*, where we use tools from computability theory to define and quantify randomness. Abandoning absolute randomness leads to the idea of "levels of randomness" that can be defined by calibrating the computability theoretic complexity of the tests we require our random sequences to pass. But, of course, following Ville's work it was not clear that even one reasonably robust level of algorithmic randomness could be defined.

Martin-Löf. This is how matters stood until 1966 and the work of Per Martin-Löf, who effectivized the notion of null set from classical measure theory and gave a satisfying definition of algorithmic randomness based on this effectivization. The basic idea is that a random sequence should not have any "rare" property, that is, that if we find a way to distinguish and describe a small collection of sequences, then no random sequence should be in our collection. The notion of null set allows us to make precise what we mean by "small."

Randomness tests like those suggested by von Mises are computable ways to narrow down which sequences can be considered random. For example, consider sequences like 0101... that have 0's in all even places. We do not want such "bad" sequences to be considered random. To test whether a sequence has this form, we take a "level-by-level" approach: Given a sequence X, we ask whether $X(0) = 0$. If so, then X fails the first level of our test. (That is, it fails to demonstrate so far that it is not a bad sequence.) Note that half of all sequences X have $X(0) = 0$, which can be formalized by saying that the set of sequences X with $X(0) = 0$ has measure $\frac{1}{2}$.

Next, we ask whether $X(0) = 0$ *and* $X(2) = 0$. If so, then X fails the second level of our test. The proportion of all sequences X that fail this second level is $\frac{1}{4}$. We continue in this fashion, testing more and more even places. A sequence X is one of our bad sequences if and only if it fails *all* levels of our test. The fact that the set T_n of sequences that fail the nth level of our test has measure 2^{-n} implies the set of bad sequences, which is the intersection of all the T_n's, has measure 0, that is, that it is what we call a *null set*.

Martin-Löf's approach was to generalize this process by considering all possible level-by-level procedures for testing randomness. We can think of such a procedure as being generated by a machine M. At each level n, this machine determines a set T_n of sequences that are deemed to have failed the test so far. It does so by enumerating strings $\sigma_0^n, \sigma_1^n, \ldots$, where we then let T_n be the collection of all sequences that begin with some σ_i^n. Of course, M needs to be fair and not, say, consider all sequences to be nonrandom, so we insist that, like in the above example, T_n contains at most a proportion 2^{-n} of all sequences (which we can formalize by saying that the measure of T_n is at most 2^{-n}). Now a sequence X *fails* M's test if it is contained in every T_n, and otherwise it *passes* this test.

We say that a sequence is *Martin-Löf random* if and only if it passes *all* such tests.[b] It can be shown that almost all sequences are Martin-Löf random (that is, that the collection of Martin-Löf random sequences has measure 1). Furthermore, Martin-Löf's notion of tests includes the ones proposed by von Mises (in the specific realization suggested by Church), the ones proposed by Ville, and indeed all "algorithmically performable" randomness tests. Thus, the objection to the idea of adding more and more specific tests as we uncover more and more Ville-like sequences is neatly circumvented.

As it turns out, Martin-Löf randomness is also quite well-behaved mathematically, and has provided a robust basis for the theory of algorithmic randomness. As Jack Lutz put it in a lecture at the *7th Conference on Computability, Complexity, and Randomness*, held in Cambridge in 2012 (in connection with work of Turing that we will discuss later), "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."

The measure of a set of sequences is the mathematical version of the probability that a sequence is in this set.

In summary, Martin-Löf reformulated all the laws that we would expect a random sequence to obey at an abstract level, based upon the idea of *effectivizing* measure theory, that is, making a computable version of measure theory. The measure of a set of sequences is the mathematical version of the probability that a sequence is in this set. Martin-Löf randomness says we regard X as random if and only if it passes each computably generated test that determines a set of computable measure 0 (as the intersection of the levels of the test). Such an X has every property that we can algorithmically describe as a set of probability 1.

Solomonoff, Kolmogorov, Levin, Chaitin, and Schnorr. There are other approaches to a definition of algorithmic randomness. For (finite) strings, a suitable definition was formulated by Kolmogorov, who argued that if a string has identifiable regularities, then we should be able to compress it, and that a compressible string should not be thought of as random. Here, we think of a machine M as a descriptional process. If an input τ is processed by M to yield an output σ, then τ is a description of σ, that is, a program that M can use to print σ. A random σ should have no short descriptions.

As an illustration, consider the sequence σ = 010101010... (1000 times). A short description τ of σ is "print 01 1000 times." This brief program produces an output of length 2000. We are exploiting the regularities of this particular σ to compress it into a short description. Kolmogorov's intuition was that for a random sequence, there should be no regularities, so that the only way to describe σ is to essentially use σ itself. More precisely, a string of length n should be random relative to some descriptional process if and only if its shortest description has length n. Like white noise, a random string should be *incompressible*.

To give a physical analogue of this idea, suppose we have a maze shaped like a binary tree of height 6, with boxes at the end. There are 2^6 possible routes to get to the boxes. One of the boxes has money in it, and someone is

b Formally, a *Martin-Löf test* is a collection S_0, S_1,... of uniformly computably enumerable sets of strings such that, if we let T_n be set of all sequences that begin with some element of S_n, then T_n has measure at most 2^{-n}. (The notion of computable enumerability is also known as recursive enumerability.) A sequence X *passes* this test if $X \notin \cap_n T_n$. A sequence is *Martin-Löf random* if it passes all Martin-Löf tests.

to tell us which. If the box is the leftmost one, all they have to say is "always turn left." If the box is to be found by say, left-right-left, this path is again easy to describe. If the place of the prize is determined randomly, though, the person would likely need to tell us the whole sequence of turns. (Li and Vitányi[27] report on an experiment of this kind about ant communication.) This compressibility approach gives rise to what is now called *Kolmogorov complexity*. For a Turing machine M, the Kolmogorov complexity $C_M(\sigma)$ of σ relative to M is the length of the shortest τ such that $M(\tau) = \sigma$. We can then take a universal Turing machine U, which can emulate any other given machine M with at most a constant increase in the size of programs, and define the (plain) Kolmogorov complexity of σ to be $C(\sigma) = C_U(\sigma)$.

A natural guess is that a sequence X is random if and only if for all n, the first n bits of X are incompressible in the sense outlined earlier. As it turns out, however, plain Kolmogorov complexity is not quite the correct notion for infinite sequences. (The reason is that in the above account, M can use more than just the bits of τ to generate σ. It can also use the length of τ, which provides an additional $\log|\tau|$ many bits of information. Using this idea, Martin-Löf showed that for *any* X, and any constant c, the plain Kolmogorov complexity of $X\restriction n$ must always dip below $n-c$ for some lengths n.)

There are several ways to modify the definition of Kolmogorov complexity to avoid this issue, the best-known being to use prefix-free codes[c] and the resulting notion of *prefix-free Kolmogorov complexity*, denoted by K in place of C. Its roots are in the work of Leonid Levin, Gregory Chaitin, and Claus-Peter Schnorr, and in a certain sense even earlier in that of Ray Solomonoff. As shown by Schnorr, it is indeed the case that X is Martin-Löf random if and only if the prefix-free Kolmogorov complexity of the first n bits of X is at least n (up to an additive constant), that is, $K(X\restriction n) \geqslant n - O(1)$.

(There are many other flavors of Kolmogorov complexity, such as time- and space-bounded ones, but C and K have been the most studied. They have a complex relationship. It is easy to show that $K(\sigma) \leqslant C(\sigma) + 2\log|\sigma| + O(1)$. Robert Solovay proved the remarkable fact that $K(\sigma) = C(\sigma) + C(C(\sigma)) + O(C^{(3)}(\sigma))$ and this result is tight in that we *cannot* extend it to $C^{(4)}(\sigma)$. There is a huge amount of research on the Kolmogorov complexity of finite strings and its applications. See, for instance, Li and Vitányi.[27])

Returning to the story of the definition of algorithmic randomness, there is another approach, developed by Schnorr, that is close in spirit to von Mises' ideas. A *martingale*[d] is a function d from strings to nonnegative reals satisfying a fairness condition:

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

We think of d as representing a betting strategy. We begin with some capital $d(\lambda)$, where λ is the empty string, and bet on the values of the successive bits of a sequence X so that the amount of money we have after n many bets is $d(X\restriction n)$. We are allowed to hedge our bets by betting some amount of our capital on 0 and the rest on 1. The displayed equation ensures that this betting is fair, that is, that the average of the returns of our bets on 0 and on 1 equals our current total. A martingale d *succeeds* on a sequence X if and only if the associated betting strategy allows us to make arbitrarily much money when betting on the bits of X, that is, $\limsup_{n\to\infty} d(X\restriction n) = \infty$. Schnorr showed that there is a notion of effective martingale such that X is Martin-Löf random if and only if no such martingale succeeds on X. This idea is close to von Mises' prediction-based approach, except that martingales allow us to spread our bets between the outcomes 0 and 1, so von Mises' intuition has a realization that works after all!

In summary, there are three basic approaches to defining random sequences: the *statistician's approach*, that a random sequence should have no computably rare properties; the *coder's approach*, that a random sequence should have no regularities that allow for compression; and the *gambler's approach*, that a random sequence should be unpredictable. In each of these cases, a natural effective realization leads to the same notion, Martin-Löf randomness.

Some Things We Have Learned

Calibrating randomness. As natural and robust as Martin-Löf's definition of algorithmic randomness is, it is only one among many reasonable notions that together allow us to calibrate levels of randomness. One way to obtain new notions of randomness is to change the collection of tests a sequence is required to pass to be considered random. For instance, we can consider Martin-Löf tests with computable measures (that is, where the measure of each level T_n is *exactly* 2^{-n}, for instance), yielding a notion called *Schnorr randomness*. Another possibility is to use martingales with different levels of effectiveness, such as ones that are computable functions from strings to nonnegative rationals, which yields a notion called *computable randomness*. Computable randomness can also be miniaturized to complexity classes, giving rise to notions such as polynomial-time randomness.

It can be shown that Martin-Löf randomness implies computable randomness, which in turn implies Schnorr randomness, and that neither of these implications can be reversed. But the separations between these notions are quite subtle, and indeed the notions coincide for sequences that are in a sense "close to computable." (More precisely, they coincide outside what are known as the *high* sequences, which resemble the Halting Problem in a certain technical sense; see Nies et al.[36]) Indeed, there is a notion of *nonmonotonic randomness*—which is like computable randomness but allows for strategies that can bet on the values of the bits of a sequence in any computable order—for which equivalence to Martin-Löf randomness is still a long-standing open question.

We can also modify our tests to yield notions stronger than Martin-Löf randomness. For instance, relaxing the condition that the nth level T_n of a Martin-Löf test must have measure at most 2^{-n}, and requiring only that the measures of the T_n's go to 0 as n goes to

c That is, descriptions that are like telephone numbers in that if τ and ρ are input descriptions to M and both give outputs, then τ is not a prefix of ρ.

d This notion is related to but distinct from that of martingale in probability theory.

infinity, yields the notion of *weak 2-randomness*, which is intermediate between Martin-Löf randomness and the notion of 2-randomness discussed below.

In some ways, weak 2-randomness is better-behaved than Martin-Löf randomness. To give an example, let us begin by considering the fact that, although almost every sequence is Martin-Löf random, it is not that easy to come up with an explicit example. That is at it should be, of course. Easily describable sequences (such as computable ones, for example) should not be random. Nevertheless, such examples do exist, the best-known being Chaitin's Ω, defined as the probability that a universal prefix-free Turing machine U halts on a given input,[e] or, more formally, as:

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

Although Ω is Martin-Löf random, it is also computationally powerful, being Turing equivalent to the Halting Problem.[f]

The existence of computationally powerful Martin-Löf random sequences is surprising, as intuitively we should expect random sequences not to contain much "useful information." (The distinction here is between the kind of information that makes a sequence hard to describe and the kind that can actually be used. If we choose 1,000 characters at random, we expect the resulting text to be difficult to describe, but would be shocked to find that it contains instructions for making a soufflé.) However, not only is it possible for a Martin-Löf random sequence to compute the Halting Problem, but by the Kučera-Gács Theorem, *every* sequence can be computed from some Martin-Löf random sequence. (See, for example, Downey and Hirschfeldt[11] for a proof.) By increasing the level of randomness, we can make these "pathological" examples disappear. If X is weakly 2-random, then it cannot compute the Halting Problem, or indeed, any noncomputable sequence that is computed by the Halting Problem, and hence in particular any noncomputable, computably enumerable set.

We do not have to go all the way to weak 2-randomness, though. There are results, beginning with work of Stephan,[38] that indicate that the Martin-Löf random sequences split into two classes: powerful ones that can compute the Halting Problem, and weaker ones that exhibit much more of the behavior we expect of random sequences, and in particular are computationally much weaker than the sequences in the first class. Franklin and Ng[17] showed that the level of randomness of these "true Martin-Löf randoms" can be captured by a natural test-based notion known as *difference randomness*. The study of notions of algorithmic randomness like this one, which are intermediate between Martin-Löf randomness and weak 2-randomness, has had an important role in recent research in the area, and helped us refine our understanding of the relationship between levels of randomness and computational power.

Another way to calibrate randomness is to relativize notions such as Martin-Löf randomness. For instance, we can consider Martin-Löf tests that are produced not by a standard Turing machine, but by a Turing machine with access to an oracle Z. If Z is the Halting Problem, for example, we obtain a notion called *2-randomness*. More generally, we have a notion of *n-randomness*, where we relativize Martin-Löf tests to the $(n-1)$st iterate of the Halting Problem.[g] Here, 1-randomness is just Martin-Löf randomness.

Much is known about this hierarchy, including some surprising facts. For example: As noted by Miller and Yu,[33] it follows from a fundamental result about Martin-Löf randomness known as van Lambalgen's Theorem (see Downey and Hirschfeldt.[11]) that if X is Martin-Löf random and is computed by an n-random sequence, then X is itself n-random. We have mentioned that we can never have $C(X\upharpoonright n) \geqslant n - O(1)$ for *all* n, but it is possible to have a sequence X such that $C(X\upharpoonright n) \geqslant n - O(1)$ for *infinitely many* n. Remarkably, Miller[30] and Nies et al.[36] showed that this condition is equivalent to 2-randomness. Miller[31] also proved a similar result saying that 2-randomness also coincides with having infinitely often maximal initial segment prefix-free Kolmogorov complexity. Indeed, it is possible to give characterizations of n-randomness for all n using unrelativized Kolmogorov complexity (see Bienvenu et al.[8]). These facts are examples of the often subtle interplay that recent research in this area has uncovered between levels of randomness, initial-segment complexity, and relative computability.

Calibrating nonrandomness. For sequences that are not Martin-Löf random, there are ways to calibrate how close they come to randomness. A natural way to do this is to consider the (prefix-free) Kolmogorov complexity of their initial segments. For example, a sequence X is *complex* if there is a computable, nondecreasing, unbounded function f such that $K(X\upharpoonright n) \geqslant f(n)$ for all n. Complex sequences can be characterized in terms of their ability to compute certain sequences that resemble the Halting Problem to some extent (see Downey and Hirschfeldt[11]), which is another example of the interplay between randomness and computability.

At the other extreme from random sequences are those with strong "anti-randomness" properties. Identifying a natural number with its binary expansion, we always have $C(\sigma) \geqslant C|\sigma| - O(1)$, because if we know a string, then we know its length. Thus, the lowest the plain Kolmogorov complexity of the initial segments of a sequence X can be is $C(X\upharpoonright n) \leqslant C(n) + O(1)$. In the 1970s, Chaitin showed that this condition holds if and only if X is computable, and asked whether the same holds for prefix-free Kolmogorov complexity. In an unpublished manuscript written in 1975, Solovay showed the surprising fact that there are noncomputable sequences X such that $K(X\upharpoonright n) \leqslant K(n) + O(1)$ for all n, though Chaitin had already shown that there are only countably many of them, and indeed that they are all computable from the Halting Problem. Such sequences are

e The value of Ω depends on the choice of U, but its basic properties do not; see Downey and Hirschfeldt.[11]

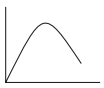
f When we say that X can be computed from Y, we mean there is a Turing machine M with an oracle tape so that if the oracle tape contains Y, then M computes X. Two objects are Turing equivalent if each can be computed from the other. Turing's Halting Problem is the classic example of a complete computably enumerable set; that is, it is itself computably enumerable, and it can compute every computably enumerable set.

g The k^{th} iterate of the Halting Problem is just the Halting Problem for Turing machines with the $(k-1)^{\text{st}}$ iterate of the Halting Problem as an oracle.

said to be *K-trivial*, and have played a major role in the theory of algorithmic randomness. For those who know some computability theory, we mention that Nies[34] showed that the *K*-trivial sequences form an ideal in the Turing degrees, and that they can be seen as giving a priority-free solution to Post's Problem (see Downey et al.[12]). Nies[34] showed that these sequences are computability-theoretically weak, and gave several characterizations of *K*-triviality in terms of randomness-theoretic weakness. For example, when we relativize the notion of Martin-Löf randomness to a noncomputable X, we expect the notion to change, as the noncomputability of X should yield some derandomization power. Nies showed that the *K*-trivial sequences are exactly those for which this intuition fails.

Many other characterizations of *K*-triviality have since been given. For example, results of Hirschfeldt et al.[21] and of Bienvenu et al.[6] show a computably enumerable set is *K*-trivial if and only if it is computed by a difference random sequence (that is, one of the "true Martin-Löf randoms" that does not compute the Halting Problem). Recent work on *K*-triviality has also revealed subclasses of the *K*-trivials that can further help us understand the fine structure of the interaction between randomness and computability.

Considering the properties of sequences with differing levels of randomness leads to the following heuristic graph, where the horizontal axis represents randomness level and the vertical axis represents maximum computational power. (One can also think that the horizontal axis represents information content, whereas the vertical axis represents maximum *useful* information content.)



Among the sequences that are neither random nor highly nonrandom are ones that can be thought of as being "partially random." For example, if Z is Martin-Löf random and we replace every other bit of Z by a 0, we obtain a new sequence Y such that $K(Y\restriction n)$ is approximately $\frac{n}{2}$. It makes sense to think of such a sequence as being "$\frac{1}{2}$-random." More generally, we can think

A remarkable feature of the theory of effective dimension is there is a tight correspondence between the classical Hausdorff dimension of a set and the effective Hausdorff dimension of its points.

of the limit behavior of the ratio $\frac{K(X\restriction n)}{n}$ as a measure of the partial randomness of a sequence X. This ratio does not necessarily have a limit, but we can look at

$$\liminf_{n\to\infty}\frac{K(X\restriction n)}{n} \quad \text{and} \quad \limsup_{n\to\infty}\frac{K(X\restriction n)}{n}$$

which both give us values between 0 and 1.

These values are also central to the theory of effective dimension. In 1919, Felix Hausdorff introduced a notion of dimension that measures the "local size" of a set in a metric space, for example, a subset of the plane. Points have dimension 0, lines have dimension 1, and the whole plane has dimension 2, but there are also objects of fractional dimension, such as well-known fractals like the Koch curve (which has Hausdorff dimension $\log_3(4)$). Starting with the work of Jack Lutz in the early 2000s, the theory of dimension has been effectivized, initially in terms of effective martingales as in Schnorr's approach to algorithmic randomness. This process has also been carried out for other notions of dimension, most notably that of packing dimension. An important fact here is that the effective Hausdorff dimension and effective packing dimension of a sequence X turn out to be exactly the liminf and limsup, respectively, in the equation explained above. Thus, these dimensions can be seen as measures of partial randomness. (See, for example, see Downey and Hirschfeldt[11] for details.)

The theory of effective dimension has also been extended to points on the plane and higher dimensional Euclidean spaces. A remarkable feature of this theory is that there is a tight correspondence between the classical Hausdorff dimension of a set and the effective Hausdorff dimension of its points. For a fairly wide class of sets $S\subseteq\mathbb{R}^n$, Hitchcock[22] showed that the Hausdorff dimension of S is the supremum of the effective Hausdorff dimensions of its individual elements, and Lutz and Lutz[28] have now given versions of this result for arbitrary sets (and for both Hausdorff and packing dimension) using relativizations of effective dimension. It is surprising that the notion of dimension, which seems so clearly to be a global property of a set, based on its "overall shape," can be

completely understood by focusing on the individual elements of the set and understanding them from a computability-theoretic perspective. This correspondence is also quite useful, and can be used to obtain new proofs and results in areas such as fractal geometry, as in Lutz and Lutz[28] and Lutz and Stull,[29] for instance.

Randomness *amplification* can be investigated in many settings. A basic question is whether (a greater degree of) randomness can always be extracted from a partially random source. In our setting, effective dimension can be used to measure the degree of randomness, and extraction can be interpreted as relative computation. One way to think of this question is that it is easy to decrease the effective dimension of a sequence in a computable way, say by changing a large proportion of its bits to 0's, but it is less clear in general whether there is a way to reverse this process.

As it turns out, the answer depends on the notion of dimension. Fortnow et al.[15] showed that if X has nonzero effective packing dimension and $\varepsilon > 0$, then there is a Y that is computable from X such that the effective packing dimension of Y is at least $1 - \varepsilon$. (In fact, they showed that Y can be chosen to be Turing equivalent to X via polynomial-time reductions, making the randomness amplification process quite efficient.) On the other hand, Miller[32] showed there is a sequence X of effective Hausdorff dimension $\frac{1}{2}$ such that if Y is computable from X, then the effective Hausdorff dimension of Y is at most $\frac{1}{2}$. (The specific value $\frac{1}{2}$ does not matter.) Greenberg and Miller[19] showed that there is a sequence of effective Hausdorff dimension 1 that does not compute any Martin-Löf random sequence. Thus, we see there are some strong senses in which randomness amplification is not possible. However, Zimand[40] showed that, remarkably, if we have *two* sequences of nonzero effective Hausdorff dimension that are sufficiently independent in a certain technical sense, then they together compute a sequence of effective Hausdorff dimension 1.

This is still an area of significant research interest. For example, we can ask about a randomness amplification process where, instead of using computable reductions, we simply seek to increase the randomness of a sequence by changing a relatively small proportion of its bits. Greenberg et al.[20] recently gave precise bounds on the proportion of bits of a sequence of effective Hausdorff dimension s that need to be changed to increase the Hausdorff dimension to a given $t > s$, in terms of the binary entropy function from information theory. They also showed that if X has effective Hausdorff dimension 1, then X can be transformed into a Martin-Löf random sequence by changing it only on the bits in a set $S \subset \mathbb{N}$ of density 0 (which means that $\lim_{n\to\infty} \frac{|S \upharpoonright n|}{n} = 0$).

It is surprising the notion of dimension, which means so clearly to be a global property of a set based on its "overall shape," can be understood by focusing on the individual elements of the set and understanding them from a computability-theoretic perspective.

Turing and absolute normality. 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, and indeed, it is known that a sequence is normal if and only if it satisfies a notion of randomness defined using certain finite-state machines much weaker than arbitrary Turing machines. Borel asked whether there are explicit examples of absolutely normal numbers. It is conjectured that e, π, and all irrational algebraic numbers, such as $\sqrt{2}$, are absolutely normal, but *none* of these have been proven to be normal to *any* base. In an unpublished manuscript, Alan Turing attacked the question of an explicit construction of an absolutely normal number by interpreting "explicit" to mean *computable*. His manuscript, entitled *A Note on Normal Numbers* and presumably written in 1938, gives the best kind of answer to date to Borel's question: an algorithm that produces an absolutely normal number.

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 tests sensitive enough to make a number normal in all bases, yet insensitive enough to allow a computable sequence to pass all such tests. We have seen that the strong law of large numbers 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."

Turing's construction remained largely unknown, because his manuscript was published only in his 1997 Collected Works.[39] The editorial notes in that volume say the proof given by Turing is inadequate and speculate the theorem could be false. Becher et al.[4] reconstructed and completed Turing's manuscript, preserving his ideas as accurately as possible while correcting minor errors. More recently, there has been a highly productive line of research connecting algorithmic randomness, computability theory, normal numbers, and approximability notions such as that of Liouville numbers; see, for instance, the papers listed at http://www-2.dc.uba.ar/profesores/becher/publications.html. Some of this work has yielded results in the classical theory of normal numbers, as in Becher et al.[3]

Some further applications. There have been several other applications of ideas related to algorithmic randomness in areas such as logic, complexity theory, analysis, and ergodic theory. Chaitin used Kolmogorov complexity to give a proof of a version of Gödel's First Incompleteness Theorem, by showing that for any sufficiently strong, computably axiomatizable, consistent theory T, there is a number c such that T cannot prove that $C(\sigma) > c$ for any given string σ.[h] More recently, Kritchman and Raz[25] used his methods to give a proof of the Second Incompleteness Theorem as well. (Their paper also includes an account of Chaitin's proof.) We can also ask about the effect of adding axioms asserting the incompressibility of certain strings in a probabilistic way. Bienvenu et al.[9] have shown that this kind of procedure does not help us to prove new interesting theorems, but that the situation changes if we take into account the size of the proofs: randomly chosen axioms can help to make proofs much shorter under a reasonable complexity-theoretic assumption like P ≠ PSPACE.

Although not central to this article, we mention there are *many* applications of Kolmogorov complexity of finite strings, for example, ones that go under the collective title of the *incompressibility method*. The idea is that algorithmically random strings should exhibit typical behavior on 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, we can compress a random input (which is now a single algorithmically random string). Chapter 6 of Li and Vitányi[27] is devoted to this technique, applying it to areas as diverse as combinatorics, formal languages, compact routing, and circuit complexity, among others. Another example is provided by the insight that the Kolmogorov complexity $C(x|y)$ of a string x *given* y *as an oracle* is an absolute measure of how complex x is in y's opinion. Historically, researchers comparing two sequences x, y of, for example, DNA, or two phylogenetic trees, or two languages have defined many distance metrics, such as "maximum parsimony" in the DNA example. But it is natural to use a measure like $\max\{C(x, y), C(y, x)\}$, if the sequences have the same length, or some normalized version if they do not. Then we know absolutely what information the strings have in common, and do not have to hand-tool a notion of distance for the application. Although C is incomputable, Vitányi and others have used computable *approximations* (such as Lempel-Ziv compression) to C to investigate general tools for understanding common information. (See, for example, Bennett et al.[5]) Another application is learning theory and *logical depth*, a notion introduced by Bennett to capture the idea that something is hard to describe in limited time. For applications to deep learning, see, for example, https://www.hectorzenil.net/publications.html.

Randomness is used in many algorithms to accelerate computations, as in the use of randomness for primality testing by Solovay and Strassen,[37] and there are problems like *polynomial identity testing*—which asks whether a polynomial in many variables is identically zero—for which there are efficient algorithms if we have a randomness source, but no known fast deterministic algorithms. It is thought that a wide class of randomized algorithms can be derandomized to yield deterministic polynomial-time algorithms, following the work of Impagliazzo and Wigderson,[23] who showed that if certain problems are as hard as we think they are, then we can provide enough randomness efficiently to derandomize problems in the complexity class BPP. Bienvenu and Downey[7] have shown that randomness can always be used to accelerate *some* computations. They showed that if X is Schnorr random, then there is a computable language L such that X can compute L (in exponential time) via a computation Φ^X (that is, a Turing machine Φ with oracle X) so that for any Turing machine M that computes L, the computation Φ^X is faster than M by more than a polynomial factor. (That is, Φ^X computes L in time f, and there are no Turing machine M and polynomial p such that M computes L in time $p \circ f$.)

Another connection with complexity theory comes from looking at the computational power of the set of random strings. There are a few reasonable ways to define what we mean by this set; one of them is to consider the strings that are incompressible in the sense of plain Kolmogorov complexity, that is $R = \{\sigma | C(\sigma) \geqslant |\sigma|\}$. It turns out to be particularly interesting to consider what sets can be reduced to this one via polynomial-time reductions. For instance, Allender et al.[1] showed that the complexity class PSPACE is contained in the collection of sets that are polynomial-time reducible to R, and other connections with complexity theory have been explored in this paper and others such as Allender et al.[2]

A particularly promising current line of research is the use of notions of algorithmic randomness to give precise, "quantitative" versions of results about almost everywhere behavior in areas such as analysis and ergodic theory, an idea that goes back to the work of

h This fact also follows by interpreting an earlier result of Barzdins; see Example 2.7.1 in Li and Vitányi,[27]

Demuth in the 1970s.[i] For example, it is a result of basic analysis that every nondecreasing function $[0, 1] \to \mathbb{R}$ is differentiable at almost every $x \in [0, 1]$ (that is, the set of x at which it is differentiable has measure 1). Brattka et al.[10] showed that the reals $x \in [0,1]$ such that every nondecreasing computable function (in the sense of computable analysis) is differentiable at x are exactly the computably random ones. Thus, computable randomness is exactly the level of randomness needed for this particular almost everywhere behavior to manifest itself. For other similar conditions, the relevant level of randomness can vary. For instance, for functions of bounded variation in place of nondecreasing ones, the corresponding level of randomness is exactly Martin-Löf randomness, as shown in Brattka et al[10] as a recasting of a result by Demuth. A source for overviews of some recent work at the intersection of algorithmic randomness with analysis and ergodic theory is the collection of slides at https://www.birs.ca/cmo-workshops/2016/16w5072/files/. Similar applications occur in physics, for instance, in studying Brownian motion (for example, by Fouché[16]) and the amount of randomness needed for quantum mechanics (for example, by Gács[18]).

Another interesting application is to the study of tilings (of the plane, say). Let $X[m, n]$ be the bits of the sequence X from positions m to n. One might think that for a Martin-Löf random X, we should have $K(X[m, n]) \geqslant n - m - O(1)$, or that at least $K(X[m, n])$ should not dip too far below $n–m$. This is not true, though, as random sequences must have long simple substrings, such as long runs of 0's. (If we know that X has infinitely many runs of 6 consecutive 0's, but only finitely many of 7 consecutive 0's, then we can make money betting on the values of the bits of X by betting that the next value is 1 each time we see six consecutive 0's.) However, for any $\varepsilon > 0$, there are *ε-shift complex* sequences X for which:

$$K(X[m, n]) \geqslant (1 - \varepsilon)(n - m) - O(1)$$

for all m and n. These sets can be coded to yield tilings with various interesting properties, such as certain kinds of pattern-avoidance. See, for instance, Durand et al.[13, 14]

Finally, randomness is thought of as "typicality" for many objects. Thus, if we wish to understand complex networks, we can try to model them using some kind of random graph. Khoussainov[24] has recently given meaning to the idea of (infinite) algorithmically random regular trees and other structures. Work is under way to adapt this idea to finite graphs and use it for practical applications.

Acknowledgments

Downey wishes to thank the Marsden Fund of New Zealand. Hirschfeldt is partially supported by NSF Grant DMS-1600543.

i Demuth came from the constructivist tradition, but independently discovered notions of randomness like Martin-Löf randomness by working on questions such as the ones discussed in this paragraph. See Kučera et al.[26] for an account.

References

1. Allender, E., Buhrman, H., Koucký, M., van Melkebeek, D., Ronneburger, D. Power from random strings. *SIAM J. Comput. 35*, 6 (2006), 1467–1493.
2. Allender, E., Friedman, L., Gasarch, W. Limits on the computational power of random strings. *Inf. Comput. 222* (2013), 80–92.
3. Becher, V., Bugeaud, Y., Slaman, T.A. On simply normal numbers to different bases. *Math. Ann. 364*, 1–2 (2016), 125–150.
4. Becher, V., Figueira, S., Picchi, R. Turing's unpublished algorithm for normal numbers. *Theor. Comput. Sci. 377*, 1–3 (2007), 126–138.
5. Bennett, C.H., Gács, P., Li, M., Vitányi, P.M.B., Zurek, W.H. Information distance. *IEEE Trans. Inf. Theory 44*, 4 (1998), 1407–1423. https://doi.org/10.1109/18.681318
6. Bienvenu, L., Day, A.R., Greenberg, N., Kučera, A., Miller, J.S., Nies, A., Turetsky, D. Computing K-trivial sets by incomplete random sets. *B. Symb. Log. 20*, 1 (2014), 80–90.
7. Bienvenu, L., Downey, R. On low for speed oracles. In *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018) (Leibniz International Proceedings in Informatics (LIPIcs))* (2018), R. Niedermeier and B. Vallée, eds. Volume 96, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Germany, 15:1–15:13.
8. Bienvenu, L., Muchnik, An. A., Shen, A., Vereshchagin, N. Limit complexities revisited. In *25th International Symposium on Theoretical Aspects of Computer Science (Leibniz International Proceedings in Informatics (LIPIcs))* (2008), S. Albers and P. Weil, eds. Volume 1, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Wadern, 73–84 (electronic).
9. Bienvenu, L., Romashchenko, A., Shen, A., Taveneaux, A., Vermeeren, S. The axiomatic power of Kolmogorov complexity. *Ann Pure Appl Logic 165*, 9 (2014), 1380–1402.
10. Brattka, V., Miller, J.S., Nies, A. Randomness and differentiability. *Trans. Amer. Math. Soc. 368*, 1 (2016), 581–605.
11. Downey, R.G., Hirschfeldt, D.R. *Algorithmic Randomness and Complexity*. Springer, New York, 2010.
12. Downey, R.G., Hirschfeldt, D.R., Nies, A., Stephan, F. Trivial reals. In *Proceedings of the 7th and 8th Asian Logic Conferences* (2003), R.G. Downey, D. Ding, S.P. Tung, Y.H. Qiu, M. Yasugi, eds. Singapore University Press and World Scientific, Singapore, 103–131.
13. Durand, B., Levin, L.A., Shen, A. Complex tilings. *J Symbolic Logic 73*, 2 (2008), 593–613.
14. Durand, B., Romashchenko, A., Shen, A. Fixed-point tile sets and their applications. *J. Comput. System Sci. 78*, 3 (2012), 731–764.
15. Fortnow, L., Hitchcock, J.M., Pavan, A., Vinochandran, V., Wang, F. Extracting Kolmogorov complexity with applications to dimension zero-one laws. In *Automata, Languages and Programming. 33rd International Colloquium, ICALP* (2006). *Venice, Italy, July 10–14, 2006. Proceedings, Part I (Lecture Notes in Computer Science)*, M. Bugliesi, B. Preneel, V. Sassone, I. Wegener, eds. Volume 4051, Springer, Berlin, 335–345.
16. Fouché, W. The descriptive complexity of Brownian motion. *Adv. Math. 155* (2000), 317–343.
17. Franklin, J.N.Y., Ng, K.M. Difference randomness. *Proc. Amer. Math. Soc. 139*, 1 (2011), 345–360.
18. Gács, P. Quantum algorithmic entropy. *J. Phys. A. Math. Gen. 34* (2001), 1–22.
19. Greenberg, N., Miller, J.S. Diagonally non-recursive functions and effective Hausdorff dimension. *B. Lond. Math. Soc. 43*, 4 (2011), 636–654.
20. Greenberg, N., Miller, J.S., Shen, A., Westrick, L.B. Dimension 1 sequences are close to randoms. *Theor. Comput. Sci. 705* (2018), 99–112.
21. Hirschfeldt, D.R., Nies, A., Stephan, F. Using random sets as oracles. *J. Lond. Math. Soc. 75* (2007), 610–622.
22. Hitchcock, J.M. Correspondence principles for effective dimensions. *Theor. Comput. Syst. 38* (2005), 559–571.
23. Impagliazzo, R., Wigderson, A. P = BPP if E requires exponential circuits: derandomizing the XOR lemma. In *STOC'97* (1999) *(El Paso, TX)*. ACM, New York, 220–229.
24. Khoussainov, B. A quest for algorithmically random infinite structures. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)* (2014). ACM, New York, Article No. 56, 9.
25. Kritchman, S., Raz, R. The surprise examination paradox and the second incompleteness theorem. *Notices Am. Math. Soc. 57*, 11 (2010), 1454–1458.
26. Kučera, A., Nies, A., Porter, C.P. Demuth's path to randomness. *B. Symb. Log. 21*, 3 (2015), 270–305.
27. Li, M., Vitányi, P. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, Berlin, 1993.
28. Lutz, J.H., Lutz, N. Algorithmic information, plane Kakeya sets, and conditional dimension. In *34th Symposium on Theoretical Aspects of Computer Science (Leibniz International Proceedings in Informatics (LIPIcs))* (2017). Volume 66, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Wadern, Article No. 53, 13.
29. Lutz, N., Stull, D.M. Bounding the dimension of points on a line. In *Theory and Applications of Models of Computation (Lecture Notes in Computer Science)*. Volume 10185, Springer, Cham, 2017, 425–439.
30. Miller, J.S. Kolmogorov random reals are 2-random. *J Symbolic Logic 69* (2004), 907–913.
31. Miller, J.S. The *K*-degrees, low for *K*-degrees, and weakly low for *K* sets. *Notre. Dame. J. Form. L. 50* (2010), 381–391.
32. Miller, J.S. Extracting information is hard: A Turing degree of non-integral effective Hausdorff dimension. *Adv. Math. 226*, 1 (2011), 373–384.
33. MillerJ.S., Yu, L. On initial segment complexity and degrees of randomness. *Trans. Amer. Math. Soc. 360* (2008), 3193–3210.
34. Nies, A. Lowness properties and randomness. *Adv. Math. 197* (2005), 274–305.
35. Nies, A. *Computability and Randomness*. Oxford Logic Guides, Volume 51, Oxford University Press, Oxford, 2009.
36. Nies, A., Stephan, F., Terwijn, S.A. Randomness, relativization, and Turing degrees. *J Symbolic Logic 70* (2005), 515–535.
37. Solovay, R., Strassen, V. A fast Monte-Carlo test for primality. *SIAM J. Comput. 6*, 1 (1977), 84–85.
38. Stephan, F. Martin-Löf random sets and PA-complete sets. In *Logic Colloquium'02 (Lecture Notes in Logic)*, Z. Chatzidakis, P. Koepke, W. Pohlers, eds. Volume 27, Association for Symbolic Logic and A K Peters, Ltd., La Jolla, CA and Wellesley, MA, 2006, 342–348.
39. Turing, A.M. *Pure Mathematics*. J.L. Britton, ed. North-Holland Publishing Co., Amsterdam, 1992
40. Zimand, M. Two sources are better than one for increasing the Kolmogorov complexity of infinite sequences. *Theor. Comput. Syst. 46* (2010), 707–722.

Rod Downey (rod.downey@vuw.ac.nz) is a professor in the School of Mathematics and Statistics, Victoria University Wellington, New Zealand.

Denis R. Hirschfeldt (drh@math.uchicago.edu) is a professor in the Department of Mathematics at the University of Chicago, IL, USA.