

## CONTRIBUTION TO “FIVE QUESTIONS IN RANDOMNESS”

JOSEPH S. MILLER

**Question 1: Why were you initially drawn to the study of computation and randomness?** It is easy to go back and find the signposts that seem to point inexorably to the present. By only telling the right parts of my story, I can make it sound like I was destined to become a computability theorist studying randomness. The truth is that the outcome was never clear to me, but there are certainly signposts worth mentioning.

My interest in computation goes back to the Texas Instruments TI-99/4A that my parents bought when I was ten. I learned to program (in BASIC) because that’s what you did with home computers back then. I think that anybody with years of programming experience has an innate understanding of what it means for something to be *computable*; I certainly don’t think I ever gave the notion—subtle in Alan Turing’s time—a moment’s thought. A function is computable if you can write a computer program to implement it. That’s all there is to it.

I got bachelor’s degrees in computer science and math; the former mostly as a precaution, but I enjoyed both programming (especially as a pastime) and theory. When I was in college I happened to read large parts of two of Gregory Chaitin’s books: “Algorithmic Information Theory” and “Information-theoretic Incompleteness”. As I recall, I stumbled across them in the library; I have no idea what I was looking for at the time. The books were quite interesting. The key insight is to define the complexity (or *information content*) of a binary string as the length of the shortest program that outputs the string. Simple strings can be compressed and complicated strings cannot. It’s an elegant idea. There are various forms of *Kolmogorov complexity*, but they are all defined in this way, with slightly different interpretations of “shortest program”. Chaitin developed the theory of (algorithmic) information and randomness, and connected it to Gödel’s incompleteness theorem, a fundamental result in the foundations of mathematics. As much as I enjoyed these themes, I never expected that I would do technical research in the field, nor even that there was an active field in which to work.

Like many in my age group, I had already been exposed to Gödel’s theorem by Douglas Hofstadter’s delightful “Gödel, Escher, Bach”. This book had a large impact on me and it’s unlikely that I would have taken a two semester graduate *Introduction to Logic* when I was in college were it not for Hofstadter. I enjoyed the course, but went to grad school certain that logic was not my subject. Despite that, I attended the logic seminar (partly because I liked the logic graduate students). Soon I was speaking in the seminar and had chosen a logician (Anil Nerode) as my advisor. Even then I meandered through a variety of possible thesis topics, finally settling on computable analysis, a hybrid of computability theory and classical

---

*Date:* September 1, 2009.

mathematics. This suited me because I was still somewhat in denial about having become a mathematical logician.

I got my degree just as many computability theorists were developing a serious interest in randomness. Rod Downey invited me to spend a year with him as a postdoc in New Zealand and I was enthusiastic because he was writing “Algorithmic Randomness and Complexity” with one of my friends from graduate school, Denis Hirschfeld. Randomness offered a rich mix of computability theory and mathematics, not entirely unlike the subject of my dissertation. I welcomed the chance to jump into what had become an active, exciting field.

**Question 2: What have we learned?** Those of us who study randomness use the word *information* in a way that is at odds with the colloquial meaning; one thing we have learned is that the two meanings are not just different, in some ways they are negatively correlated. First, we must understand the difference. As an analogy, say you tuned your radio to static. After ten minutes of listening to white noise, it is highly unlikely that you would have learned anything about current events, why dolphins don’t drown when they sleep,<sup>1</sup> or any other subject. If you want to learn something, you would do better listening to ten minutes of NPR. Colloquially, we would say that NPR contains information and radio static doesn’t. To distinguish it from the other notion, let’s call this kind of information “useful information”.

Now assume that you have recorded the two ten minute segments as compressed audio files.<sup>2</sup> The radio static should be much *less* compressible than the ten minutes of NPR. Why? Because it is random noise and compression is only possible when there are patterns to exploit. So, in the sense of Kolmogorov complexity, the white noise contains *more* information.

Let’s turn to the mathematical context, drawing on our analogy to help understand the distinction between the two notions of information. Consider an infinite binary sequence  $X$ . For example, we might have  $X = 00100100001111110\dots$ , continuing without end and perhaps with no discernible pattern. In computability theory, our measure of “useful information” is the *computational power* of  $X$ , in other words, what can be computed *from*  $X$ . Say that we have the binary sequence  $X$  written on an (infinite) hard drive. With this drive plugged into our computer, we can write programs that have access to  $X$ . If  $X$  is computable, these programs can’t do anything that we couldn’t do without the information on the hard drive.<sup>3</sup> But if  $X$  is not computable, we can write programs that do new things. As a trivial example, there is a program that computes  $X$  by simply copying it off the drive. More generally, it is possible that, using  $X$ , we can compute another sequence  $Y$  (which may not, itself, be computable). In that case we say that  $Y$  is *computable*

---

<sup>1</sup>This is apparently because only half of a dolphin’s brain is asleep at any given time.

<sup>2</sup>The most commonly used forms of audio compression do not perfectly preserve the uncompressed audio. By making changes that are (ideally) not detectable by the listener, they allow for much more efficient compression. For our example it is best to assume we are using *lossless* compression. The fact that white noise contains so little information in the colloquial sense is closely related to the fact that we cannot perceptually distinguish two similar white noise samples. This in turn allows for efficient compression of white noise if all we want to do is maintain the perceptual experience.

<sup>3</sup>It is entirely possible that access to a computable sequence  $X$  would allow us to perform some computations much faster than would otherwise be possible, but we are not interested in the efficiency of computation here; a program that won’t halt until well after the sun has burnt itself out is still a perfectly good program from our perspective.

from  $X$  or that  $Y$  is *Turing reducible* to  $X$ . It is also possible, for example, that  $X$  computes a function that grows faster than every computable function. Such sequences form an important class in computability theory. There are plenty of other ways that  $X$  could be computationally useful, in each case allowing us to do something that we couldn’t do without access to the hard drive containing  $X$ .

Next we want to capture the randomness-theoretic information contained in  $X$ . For that we use the Kolmogorov complexity of its (finite) initial segments. As was already mentioned, the Kolmogorov complexity of a string is the length of the most compressed form of the string. Think of it as the length of the irreducible part; what’s left when all redundancy and pattern is removed. Hopefully, you see that it makes sense to call this *information*, even though it is quite different from what we meant by “information” in the previous paragraph.

If none of the initial segments of  $X$  can be compressed (beyond some fixed amount and using an appropriate variation of Kolmogorov complexity), we say that  $X$  is *Martin-Löf random*.<sup>4</sup> By this definition, a random sequence has a lot of information, in the sense that its initial segments cannot be generated by programs shorter than themselves. This is an elegant definition, and useful if we want to talk about the two meanings given to “information”. It may not be immediately clear what it has to do with our intuitive notion of randomness, which has more to do with flipping coins and rolling dice. However, a Martin-Löf random sequence is guaranteed to “look like” a sequence you would get by flipping a coin and assigning 1 to heads and 0 to tails. It is worth a digression to understand the connection. If you flip a coin 1,000 times, you would expect roughly 500 heads. How can this be if every possible sequence of heads and tails is equally likely? The reason is that the vast majority of sequences of 1,000 heads and tails have roughly 500 heads, so the probability that the sequence you generate will have this property is very high. Just to put numbers to this observation, if you tried this experiment one trillion times, you would expect to see fewer than 400 or more than 600 heads in only about 180 of those trials! What does this have to do with compression? Since there are comparatively few sequences of length 1,000 with fewer than 400 or more than 600 ones, each of these can be assigned as the output to a relatively short program. Hence, these pathological sequences have Kolmogorov complexity less than their lengths. So a Martin-Löf random sequence will, if you keep a running count, have roughly the same number of ones as zeros. Other properties that you would expect a randomly generated sequence—one generated by flipping a fair coin—to have can be analyzed in the same way. The point is that Martin-Löf random sequences look random in exactly the way you would hope.

Now that they have both been described, it should be clear that randomness-theoretic information and computability-theoretic information are very different notions. But remember our goal is to show that the two meanings are not just different, but at odds with each other. Let us return to our radio analogy. Taking Martin-Löf random sequences as our analog of white noise and using the computational power of  $X$  as a measure of its useful-information, how well does our analogy fit the formal situation? Not as well as we might hope; Péter Gács and Antonín Kučera independently proved that there is no limit on the computational power

---

<sup>4</sup>This is not actually how Per Martin-Löf defined his notion of randomness in 1966. It is an approach to the same notion of randomness that was developed by Leonid Levin, Gregory Chaitin and Claus-Peter Schnorr in the seventies.

of Martin-Löf random sequences. In other words, whatever you want to compute, there is a random sequences that computes it, so Martin-Löf randoms *can* contain useful information. One weakness in the analogy is that we are allowing ourselves to work much harder to extract this information than one is willing to work when listening to the radio. The bigger weakness—one that will let us rescue our analogy when it is resolved—is that a Martin-Löf random sequence  $X$  is not “absolutely random”, whatever that might mean, but only random to the extent that computer programs cannot exploit patterns in  $X$  to compress its initial segments. What we will argue is that the more random a sequence is, the less computationally useful, *and conversely* the less computationally useful a random sequence is, the more random. But for this we have to understand what we might mean by “more random”.

One way to strengthen the notion of Martin-Löf randomness is to draw on our earlier discussion of computability theory. Say that our programs have access to a sequence  $Z$  written on an (infinite) hard drive. If we cannot compress the initial segments of a sequence  $X$  *using*  $Z$ , then we say that  $X$  is Martin-Löf random *relative to*  $Z$ . This gives us a stronger notion of randomness, depending on the computational power of  $Z$ . This is not the only way that we strengthen Martin-Löf randomness in practice, but it will be enough to make our point.

We can now give examples illustrating the negative correlation between the two types of information. First, assume that  $X$  computes  $Y$  and that both sequences are Martin-Löf random. This assumption places an *upper* bound on the computability-theoretic information in  $Y$ : it cannot be more computationally useful than  $X$  because anything that  $Y$  computes,  $X$  also computes. On the other hand, it turns out that  $Y$  inherits  $X$ ’s randomness in the sense that if  $X$  is random relative to  $Z$ , then so is  $Y$ . This can be seen as a *lower* bound on the degree of randomness of  $Y$ : any  $Z$  that compresses  $Y$  must also compress  $X$ . So we have evidence supporting the claim that the less computationally useful a random sequence is, the more random.

For the other direction, assume that the Kolmogorov complexity of the initial segments of  $X$  always exceeds that of  $Y$ , and again that both are random. In this case we have placed an *upper* bound on the randomness-theoretic information in  $Y$ . Do we get a corresponding *lower* bound on the computability-theoretic information? We do; it turns out that  $Y$  compresses every sequence that  $X$  compresses. In other words,  $Y$  is at least as useful in exploiting patterns in other sequences as  $X$  is. Thus we have evidence that the less random a sequence is, the more computationally useful. There are many other technical results that flesh out our claim that randomness-theoretic and computability-theoretic information are not just very different notions, but for sufficiently random sequences, inversely related.

Something very interesting happens at the other extreme. The sequences that have the lowest possible randomness-theoretic information turn out to be exactly the sequences that are computationally useless in a specific sense. In other words, if we look all the way at the low information end of the spectrum, the two types of information seem to coincide. This should make more sense if we return to our radio analogy. Consider a ten minute long pure tone. Like white noise, this is bad programming and contains no *useful* information. On the other hand, if you were to record this segment as a compressed audio file, you would find that it takes up very little space, much less even than the ten minute segment recorded from NPR. So it contains very little information, no matter which meaning we use.

Let’s turn to the mathematical context. We say that  $X$  is *K-trivial* if each initial segment of  $X$  has the lowest possible Kolmogorov complexity of any finite string of that length (up to a fixed tolerance and using an appropriate variation of Kolmogorov complexity). These are the sequences with no randomness-theoretic information. We have already identified the sequences with absolutely no computability-theoretic information; these are just the computable sequences. It isn’t hard to see that every computable sequence is *K-trivial*. However, Robert Solovay constructed a noncomputable *K-trivial* sequence. Once again, we have to be careful to rescue our analogy. We need to restrict our attention to a specific kind of computation. We say that  $X$  is *low-for-random* if every Martin-Löf random sequence is Martin-Löf random *relative to X*. In other words, having  $X$  doesn’t let you compress anything that isn’t already compressible. While a low-for-random sequence might be useful for certain types of computations, they are assumed to be useless in this one specific sense.<sup>5</sup> A beautiful and deep result of André Nies (building on the work of others) states that the *K-trivial* and the *low-for-random* sequences are the same. Being computationally useless in the sense of not aiding compression is the same as being maximally compressible. On the low end of the spectrum, the two notions of information align nicely.

**Question 3: What don’t we know (yet)?** There is a great deal that we don’t know. Some of our ignorance can be stated in the form of precise technical questions (see Question 4). Answers to those questions may impact our intuitive understanding of randomness and reveal unexpected patterns. On the other hand, these high level patterns and revised intuitions direct our technical research and lead to precise questions. I want to focus on the difficulty of this latter process, on the patterns and intuitions that we don’t yet know how to formalize. Some of the more intriguing areas of our ignorance are those that we haven’t been able to translate into explicit mathematical questions, let alone answer.

There are a number of interesting high level patterns that have emerged in our technical results for which we haven’t found a unifying explanation. For example, there are several other characterizations of the *low-for-random* sequences, beyond the two that have been mentioned. This is a common feature of natural mathematical notions, that they have many, often very different looking, characterizations. All of the different ways we have found to characterize the low-for-random sequences have something in common: they refer to Kolmogorov complexity, to Martin-Löf randomness, or to convergent series (which can be viewed as a hidden reference to Kolmogorov complexity). Unlike properties that have similar definitions, low-for-random seems to have no purely computability-theoretic characterization. However, it is not clear how you could formulate a result to capture this observation, or even what should be meant by “purely computability-theoretic”. The difficulty inherent in satisfactorily formalizing intuitive notions like “purely computability-theoretic”—especially when it’s not clear that experts would agree on the intuition—is an obstacle to translating our ignorance into precise questions.

Another example, one that we have already come across, is that it is not clear what it should mean for one sequence to be *more random* than another. In the answer to Question 2, we saw evidence supporting the claim that there is a negative

---

<sup>5</sup>Although a low-for-random sequence need not be computable, there are very strict limitations on its computational usefulness. For example, it cannot compute a function that grows faster than every computable function.

correlation between the two types of information, that the more random a sequence is, the less computationally useful, and the less computationally useful a random sequence is, the more random. This is a higher level pattern with a lot of technical evidence to support it (and a compelling analogy!), but we don't know how to formulate a result that ties all of this evidence together into a coherent picture. Part of the reason is that we don't have an agreed upon precise formulation of "more random". If the Kolmogorov complexity of the initial segments of  $X$  always exceeds that of  $Y$ , then it seems reasonable to say that  $X$  is more random than  $Y$ . Similarly, if  $X$  is random relative to  $Z$  whenever  $Y$  is random relative to  $Z$  (i.e., any  $Z$  than can compress  $X$  can also compress  $Y$ ), then it is reasonable to say that  $X$  is more random than  $Y$ . These are the two ways we interpreted "more random" in the answer to Question 2, and it is worth pointing out that for Martin-Löf random sequences, the first implies the second. But there are reasons to believe that neither relationship captures the intuition that  $X$  is more random than  $Y$ . Even the second condition seems to be too demanding.

It may be the case that intuitive notions like "purely computability-theoretic" or "more random" are too vague to support formalization, and that the higher level patterns we would like to capture are only partial truths, allowing no unifying explanation. The point is that a large part of our ignorance lies outside of the long lists of technical question that we haven't yet answered.

**Question 4: What are the most important open problems in the field?**

I will discuss just one problem. There is no shortage of open problems in the intersection of computability theory and randomness,<sup>6</sup> but this one has survived focused investigation since it was formulated in 1998 by Muchnik, Semenov and Uspensky, which gives it seniority in a young and fast-moving field. Age alone does not make it important, of course.

Up until now we have focused exclusively on Martin-Löf randomness. There are good reasons why it is the most widely studied notion of its type, but it is not the only attempt to define randomness for a sequence using the idea that computer programs are too weak to find any pattern in the sequence, nor is it the easiest to explain. In 1971, Claus-Peter Schnorr used programs to bet against sequences, rooting his definition in the unpredictability of randomness. A program betting on a sequence  $X$  starts with \$1 and bets on one bit at a time (remembering the outcomes of previous bets but obviously not being allowed to look ahead). If there is no upper bound on the amount of money the program reaches while betting against  $X$ , we say it *wins against*  $X$ . If no betting program wins against  $X$ , then  $X$  is called *computably random*. As a simple example, assume that every other bit of  $X$  is zero. We can easily write a program that bets nothing on the odd bits and bets all of its capital that each even bit is a zero. This program wins against  $X$ , so  $X$  is not computably random. This notion of randomness is weaker than Martin-Löf's in the sense that every Martin-Löf random sequence is computably random but the reverse is false.<sup>7</sup>

<sup>6</sup>André Nies and I published a list of open problems, many of which remain open (Joseph S. Miller and André Nies. Randomness and computability: open questions. *Bulletin of Symbolic Logic*, 12(3):390–410, 2006).

<sup>7</sup>It has been shown that any sequence that is computably random but not Martin-Löf random computes a function that grows faster than every computable function. In other words, assuming an *upper* limit on the randomness of a somewhat random sequence puts a *lower* limit on its

Schnorr also gave a characterization of Martin-Löf randomness in terms of betting strategies, but these strategies went somewhat beyond what is possible for betting programs. For this reason, Schnorr criticized Martin-Löf’s definition of randomness as too restrictive. However, it is possible that with a slight change in the kind of bets we let our programs make, Martin-Löf randomness might be characterized in a way more in line with computable randomness. This is the context in which we should understand Muchnik, Semenov and Uspensky’s question.

Say that we allow a program to bet on the bits of  $X$  in *any order*. At each step in the betting game, the program is allowed to choose an amount of its current capital *and* a position of  $X$  that hasn’t been chosen before (hence hasn’t been revealed). It could begin by betting \$0.50 that the 17th position is a zero. Its second bet is conditional on the outcome of the first. For example, if it wins, it might bet \$1.25 that the 3rd position is a one, but if it loses, bet \$0.25 that the 51st position is a zero. We call this a *non-monotonic* betting strategy. Once again, we say that the program *wins* if there is no upper bound on the amount of money the program reaches while betting against  $X$ . If no non-monotonic betting program wins against  $X$ , then  $X$  is called *non-monotonically random*. The open question is simple: is this randomness notion exactly the same as Martin-Löf randomness? It is known that it is quite a bit stronger than computable randomness and that every Martin-Löf random sequence is non-monotonically random. If they turn out to be the same it would be a striking response to Schnorr’s criticism.

**Question 5: What are the prospects for progress?** I would like to see greater integration of tools from probability and measure theory, the branches of mathematics that classically explore the notion of randomness. I think it is entirely possible that some of our open problems, such as the one discussed in Question 4, will be answered using more sophisticated ideas from classical mathematics. These ideas can also spawn new research directions. For example, Bjørn Kjos-Hanssen and Willem Fouché have studied the analog of Martin-Löf randomness for continuous functions using ideas that probability theorists developed to model and study Brownian motion. I would also like to see tools from the study of computability-theoretic randomness applied to measure theory. There are few examples of this as of yet, but one was recently given by Jan Reimann, who used these tools to give a new proof of Frostman’s Lemma, a result from fractal geometry. Overall, I see the integration of classical mathematics into the study of Kolmogorov complexity and randomness to be an important part of the long term development of the field.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF WISCONSIN, MADISON, WI 53706-1388, USA  
*E-mail address:* `jmiller@math.wisc.edu`

---

computational power (i.e., useful information). This gives us another example of the negative correlation between the two notions of information that we discussed in Question 2.