Brit. J. Phil. Sci. **51** (2000), 255–271

The Reliability of Randomized Algorithms

Don Fallis

ABSTRACT

Recently, certain philosophers of mathematics (Fallis [1997]; Womack and Farach [1997]) have argued that there are no *epistemic* considerations that should stop mathematicians from using probabilistic methods to establish that mathematical propositions are true. However, mathematicians clearly should not use methods that are *unreliable*. Unfortunately, due to the fact that randomized algorithms are not really random in practice, there is reason to doubt their reliability. In this paper, I analyze the prospects for establishing that randomized algorithms are reliable. I end by arguing that it would be inconsistent for mathematicians to suspend judgement on the truth of mathematical propositions on the basis of worries about the reliability of randomized algorithms.

1 Randomized algorithms

It is very time consuming to perform certain computational tasks—such as determining whether or not a very large number is prime—using deterministic algorithms. As a result, computer scientists have developed a large number of *randomized* algorithms. A randomized—or probabilistic—algorithm is an algorithm that makes several random choices during a computation. For example, a randomized algorithm might pick numbers at random—from a finite set of numbers—in order to determine exactly which calculations to perform. According to the computer scientists Rajeev Motwani and Prabhakar Raghavan, 'for many applications, a randomized algorithm is the simplest algorithm available, or the fastest, or both' ([1995], p. ix).

In order to take advantage of the speed of randomized algorithms, however, we have to be willing to give up a little bit of certainty. Randomized algorithms give the right answer most of the time, but they are not guaranteed to give

the right answer all of the time.[1] A case in point is the randomized algorithm—one of the earliest—developed by the computer scientist Michael Rabin ([1980]) to determine whether or not a number is *prime*.

Rabin's algorithm is based on the following result: if a number *n* is *composite*—that is, if *n* is not prime, then over three-quarters of the numbers between 1 and *n* are *witnesses* to the compositeness of *n*. That is, most of the numbers between 1 and *n* have a property—which can be tested for very quickly—that implies that *n* is composite. Rabin's algorithm consists of the following steps: first, a whole bunch of numbers between 1 and *n* are picked at random. Second, each of these numbers is tested to determine if any of them is a witness. If any of the numbers is a witness, then the algorithm reports that *n* is composite. If none of the numbers are witnesses, then the algorithm reports that *n* is prime. If *n* is composite, the overwhelming odds are that at least one of the numbers picked at random will be a witness to the compositeness of *n*. As a result, if none of the numbers are witnesses, the overwhelming odds are that *n* is prime. Even so, if none of the numbers are witnesses, it is not absolutely certain that *n* is prime.[2]

In the twenty years since Rabin developed his algorithm, randomized algorithms have been developed to perform a whole variety of computational tasks. Randomized algorithms have mainly been used to achieve practical ends such as efficient resource allocation and secure communication (see Motwani and Raghavan [1995]). However, they have also been used to achieve purely theoretical ends. *Monte Carlo simulations* are randomized algorithms that are used to simulate complex physical processes and thereby confirm certain scientific hypotheses. For example, A. M. Ferrenberg, D. P. Landau, and Y. J. Wong ([1992]) used a Monte Carlo simulation to determine the critical temperature at which certain materials become magnetized. In addition, Monte Carlo simulations have even been used to confirm certain *philosophical* hypotheses (see Skyrms [1996], pp. 15–6, 107).

Randomized algorithms might also be used to establish that certain mathematical propositions are true. For example, Rabin's algorithm has been used to show that $2^{400}-593$ is the largest prime below 2^{400} (see Rabin [1980], p. 136). However, while the use of randomized algorithms is accepted in scientific research, it is clearly not accepted in mathematical research. As the computer scientist David Harel puts it, 'as long as we use probabilistic algorithms only for petty, down-to-earth matters such as wealth,

[1] Actually, there are two different types of randomized algorithm. As the computer scientist David Harel puts it, 'those that are always fast and probably correct are dubbed **Monte Carlo**, and those that are always correct and probably fast are termed **Las Vegas**' ([1989], p. 280). In this paper, I will only be concerned with Monte Carlo algorithms.

[2] Another randomized algorithm for primality testing was developed at about the same time (see Solovay and Strassen [1977]). It differs from Rabin's algorithm only with respect to the precise definition of a witness.

health, and survival, we can easily make do with very-likely-to-be-correct answers to our questions. The same, it seems, cannot be said for our quest for absolute mathematical truth' ([1989], p. 295).

Recently, a few mathematicians have disputed this claim. For example, the mathematician Reuben Hersh suggests that mathematicians should use 'machine computation, numerical evidence, probabilistic algorithms, if we find them advantageous' ([1997], p. 59). In a similar vein, the mathematician Doron Zeilberger claims that mathematicians should be willing to forgo the certainty of deductive proof 'since ''almost certainty'' can be bought so much cheaper' ([1993], p. 980).

A couple of things need to be established, however, before we can safely conclude that executing a randomized algorithm is a legitimate form of mathematical justification. First, it needs to be established that executing an algorithm that only provides 'almost certainty' is a legitimate form of mathematical justification.[3] Recently, certain philosophers (Fallis [1997]; Womack and Farach [1997]) have argued—against Harel—that it is. Second, it needs to be established that randomized algorithms really do provide 'almost certainty'. In other words, it needs to be established that randomized algorithms are reliable. In this paper, I analyze the prospects for establishing that randomized algorithms are reliable. I end by arguing that it would be inconsistent for mathematicians and scientists to doubt the reliability of randomized algorithms.

2 Proofs of reliability

At first glance, establishing that randomized algorithms are reliable might not appear to be a very difficult task. For example, it should be clear from the description given above that Rabin's algorithm is very reliable. In fact, for most randomized algorithms, there is a *proof* that the algorithm is reliable. Unfortunately, these proofs of reliability are based on the assumption that randomized algorithms actually pick numbers *at random*. However, in practice, randomized algorithms definitely do not pick numbers at random. According to the computer scientist Eric Bach, 'when a randomized algorithm is implemented, one always uses a sequence whose later values come from earlier ones in a *deterministic* fashion. This invalidates the assumption of independence and might cause one to regard results about probabilistic algorithms with suspicion' (Bach [1991], p. 30).

[3] Since randomized algorithms are almost invariably executed on digital computers, it also needs to be established that executing an algorithm on a digital computer can be a legitimate form of mathematical justification. In the wake of the *computer* proof of the four-color theorem, a number of philosophers argued—successfully, I think—that executing algorithms (at least deterministic ones) on a computer can be legitimate (see e.g. Detlefsen and Lukar [1980] and Teller [1980]).

When an algorithm is supposed to pick numbers at random, computer programmers typically use a deterministic algorithm known as a *random number generator* (RNG). An RNG produces a sequence of *pseudorandom* numbers that appear to have been chosen at random—from a given interval of the natural numbers. In other words, a sequence of pseudorandom numbers exhibits a high degree of disorder. For example, every number—as well as every sequence of numbers—appears with approximately equal frequency in a sequence of pseudorandom numbers.

The most common type of RNG is the *linear congruential generator* (LCG). An LCG produces a sequence of numbers $<a_1, a_2, \ldots, a_{r-1}, a_r>$ where the value of each number in the sequence is calculated using the formula $a_{n+1} = (ba_n + c) \bmod d$. a_1 is known as the *seed* and a new value for a_1 is chosen each time the LCG is executed. b, c, and d are constants that define the particular LCG. d is known as the *modulus* of the LCG and is typically a very large prime number.

Even though a sequence of numbers produced by an RNG exhibits a high degree of disorder, it is still not a sequence of numbers chosen at random (see e.g. Ekeland [1993], pp. 18–27). In a sequence of numbers produced by a random process, the value of every number in the sequence is completely independent of the values of previous numbers in the sequence. In a sequence of numbers produced by an RNG, however, every number is completely determined by previous numbers in the sequence. For example, every number is completely determined by the preceding number in a sequence produced by an LCG.

Since, *in practice*, randomized algorithms do not pick numbers at random, the aforementioned proofs of reliability—though they are valid arguments—appeal to a false premise.[4] An argument—even a valid one—that appeals to a false premise does not establish the truth of its conclusion—in this case, that randomized algorithms are reliable *in practice*. Thus, the aforementioned proofs of reliability do not serve to establish that randomized algorithms actually provide 'almost certainty'.

At this point, we have essentially two options if we want to establish that randomized algorithms are reliable. First, we can find some other evidence for the reliability of randomized algorithms implemented with RNGs. Second, we can find a way for randomized algorithms to pick numbers at random—i.e.

[4] It has been suggested that randomized algorithms are superior to many other probabilistic methods (such as the probabilistic DNA proof discussed in Fallis [1997]) because it is possible to calculate *precisely* the probability that a randomized algorithm will give an incorrect answer. However, the probability value generated by such a calculation is only accurate if the randomized algorithm really picks numbers at random (and if the possibility of hardware failures is ignored). Since, *in practice*, randomized algorithms do not pick numbers at random, it is not so easy to calculate precisely the probability that a randomized algorithm will give an incorrect answer.

we can design randomized algorithms for which the problematic premise is true. In the following sections, I will examine these two options with regard to Rabin's algorithm. Essentially the same issues arise, however, for any randomized algorithm.

3 Empirical evidence of reliability

One obvious way to establish that a randomized algorithm implemented with an RNG is reliable is to subject the algorithm to some empirical tests. For example, we might check that the randomized algorithm gives the right answer in those cases where we already know what the right answer is. In fact, Rabin gathered this sort of empirical evidence for the reliability of his algorithm.[5] Rabin executed his algorithm on a bunch of numbers for which he already knew the right answer—namely, the Mersenne numbers less than 2^{500} (see Rabin [1980], p. 136). The algorithm distinguished the primes from the composites with 100% accuracy.

Empirical tests certainly provide some evidence for the reliability of a randomized algorithm. Indeed, this sort of empirical evidence for reliability might be sufficient if we are only using randomized algorithms for 'petty, down-to-earth matters such as wealth, health, and survival'. However, if we are using randomized algorithms to achieve theoretical ends—e.g. to establish that $2^{400} - 593$ is the largest prime below 2^{400}, then we are subject to higher standards of evidence.

In pure mathematical—and scientific—research, we want to have compelling evidence for the conclusions that we draw. For instance, a mathematical proof provides compelling evidence that a theorem is true; anyone who understands the proof is compelled to believe that the conclusion follows from the premises. Similarly, if it has been proven that a particular deterministic algorithm for primality testing always gives the right answer, then executing this algorithm provides compelling evidence that a number is prime.[6]

Unfortunately, the fact that Rabin's algorithm has worked for a small finite number—i.e. measure zero—of the infinitely many natural numbers is just not very compelling evidence for the reliability of the algorithm. After all, the mere fact that the Goldbach conjecture has been verified to hold up to 4.10^{11} (see Sinisalo [1993]) does not lead many mathematicians to believe that the conjecture holds for all natural numbers.

[5] Interestingly enough, Rabin did not even mention the fact that his algorithm was implemented with an RNG (and, thus, did not really pick numbers at random). This seems to suggest that he had little doubt about the reliability of his algorithm implemented with an RNG. Whether or not he had good grounds for believing that it was reliable, however, is what is at issue.

[6] Compelling evidence is, of course, often available in science as well as in mathematics. For example, the predicted return of Halley's comet in 1758 provided fairly compelling evidence for Newton's law of universal gravitation.

In addition to performing empirical tests, we can take the advice of the computer scientist Donald Knuth and 'run each Monte Carlo program at least twice using quite different sources of random numbers, before taking the answers of the program seriously' ([1981], p. 156). However, this will not provide us with very compelling evidence of reliability either.

In the absence of compelling evidence, we ought to suspend judgement on mathematical and scientific claims. As the Isaac Levi puts it, 'the need to take decisions [. . .] in pure research [. . .] does not mandate or even excuse unjustified resolution of conflict or leaping to conclusions' ([1984], p. 270). In particular, if there is not compelling evidence for the reliability of Rabin's algorithm, mathematicians should suspend judgment, for example, on whether or not $2^{400} - 593$ really is the largest prime below 2^{400}. As a result, what we need is evidence for the reliability of Rabin's algorithm that is so compelling that it would not be legitimate for mathematicians to suspend judgement on the truth of mathematical propositions that Rabin's algorithm has been used to establish.[7]

4 Proofs of reliability revisited

One way in which we might try to get compelling evidence is to *prove* that a randomized algorithm implemented with an RNG is reliable. In particular, we might attempt to give an argument that is similar to the aforementioned proofs of reliability, but that does not appeal to the assumption that the randomized algorithm actually picks numbers *at random*. Making such an attempt requires us to look more carefully at exactly how Rabin's algorithm works.

Let us say that m is a *liar* if n is composite, $1 \leq m \leq n$, and m is *not* a witness to the compositeness of n. Rabin's algorithm chooses a finite sequence of numbers $<a_1, a_2, \ldots, a_{r-1}, a_r>$ such that for all $i \leq r$, $1 \leq a_i \leq n$. Let us say that the sequence is a *bad sequence* if for all $i \leq r$, a_i is a liar. If a bad sequence of numbers is chosen, then Rabin's algorithm will incorrectly report that a composite number is prime. However, if bad sequences are unlikely to be chosen, then Rabin's algorithm is likely to be correct whenever it reports that n is prime. In other words, if bad sequences are unlikely to be chosen, then Rabin's algorithm is reliable.[8]

[7] The problem here is not that the mathematician necessarily has specific evidence that Rabin's algorithm is unreliable. The problem is that she may not have very compelling evidence that it is reliable.

[8] The mathematician Carl Pomerance worries that 'if we happen to recognize the set $\{b_1, \ldots, b_{100}\}$ for having an inordinately large number of strong pseudoprimes, then there may be a higher chance n is composite' ([1981], p. 100). In other words, if we have reason to believe that a bad sequence has been chosen in a particular case, then we should lower our confidence in the primality of n. This is certainly true. However, as long as bad sequences are unlikely to be chosen, the algorithm itself is reliable.

Because witnesses are very numerous, bad sequences are very rare. More precisely, at most $1/4^r$ of the sequences of length r are bad sequences. For example, less than one percent of the sequences of length 4 are bad sequences. Since bad sequences are rare and since a truly random process has an equal chance of choosing any particular sequence from a set of sequences, it is unlikely that a truly random process will choose a bad sequence. In fact, the longer the sequence that is chosen, the more unlikely it is that a truly random process will choose a bad sequence.

Even so, using a truly random process might not be the only way to insure that bad sequences are unlikely to be chosen. For instance, even though RNGs do not choose sequences at random, it might nevertheless be the case that certain RNGs are unlikely to choose bad sequences. Thus, in order to establish the reliability of Rabin's algorithm implemented with an RNG, we can try to show that the RNG in question is unlikely to produce bad sequences.

There are essentially two things that make it difficult to establish that an RNG is unlikely to choose bad sequences. First, any given RNG is capable of producing only a very small subset of the sequences of length r. There are n^r sequences of length r such that for all $i \leq r$, $1 \leq a_i \leq n$, but a particular LCG can only produce n sequences of length r—i.e. one for each possible value of a_1. So, even though there are very few bad sequences of length r, bad sequences might be very prevalent in this small subset. In fact, it would be easy to construct a deterministic algorithm that only produces bad sequences. Second, even if there are very few bad sequences in the small subset, it is not necessarily unlikely that a bad sequence will be chosen. For example, the particular sequence that is produced by an LCG is completely determined by the value of a_1. Whether or not a bad sequence is unlikely to be chosen depends upon how this value is chosen and it is certainly possible to choose it in such a way that the bias is toward the few bad sequences in the small subset.

Computer scientists have made a number of recommendations for how to address the first worry—i.e. that bad sequences might be prevalent in the small subset of sequences that an RNG is capable of producing. First, a few computer scientists recommend that our randomized algorithms use RNGs that combine several different types of RNG—such as the LCG, the 1/P generator, the x^2 mod N generator, etc. It turns out, however, that it is very difficult to do better than a simple LCG and it is very easy to do worse (see Park and Miller [1988]). As the mathematician B.D. Ripley points out, 'we know very little about such combination generators [. . .] This may well be an excellent generator, but to my knowledge none can prove so' ([1988], p. 55). In any case, there is no a priori reason to believe that combination generators produce fewer bad sequences than a simple LCG.

Second, many computer scientists recommend that our randomized algorithms only use 'high quality' RNGs—i.e. ones that pass a number of statistical

tests of randomness. I should remark at this point that there is an ambiguity in the use of the term *random*. On the one hand, it is typically used to refer to a process that has an equal chance of choosing any particular sequence of numbers from a set of sequences. This is the property of random processes that the proofs of reliability appeal to. On the other hand, it is also used to refer to certain sequences of numbers themselves. Basically, a 'random' sequence is a sequence that exhibits a very high degree of disorder (see e.g. Knuth [1981], pp. 127–51). A 'high quality' RNG is an RNG that only produces 'random' sequences.

Unfortunately, the mere fact that an RNG only produces 'random' sequences does not give us reason to believe that the RNG produces very few bad sequences. Rabin established that there are not many liars, but this does not give us any information about exactly how the liars are distributed between 1 and *n*. As far as we know, sequences of liars—i.e. bad sequences—might exhibit a very high degree of disorder. As a result, there is no *a priori* reason to believe that a 'random' sequence is unlikely to be a bad sequence. In fact, according to A. M. Ferrenberg, D. P. Landau, and Y. J. Wong, even 'ostensibly high quality random number generators may lead to subtle, but *dramatic*, systematic errors for some [randomized] algorithms' ([1992], p. 3384).

Third, some computer scientists recommend that we simply prove that bad sequences for *a particular randomized algorithm* are not prevalent in the subset of sequences that *a particular RNG* is capable of producing. The computer scientist Eric Bach has done essentially this for Rabin's algorithm and a few LCGs. More precisely, Bach ([1991], p. 50) proved that if a_1 is picked at random—from the interval 1 to *n*—and a sufficiently long sequence is produced by the LCG, then it is unlikely to be a bad sequence for Rabin's algorithm. As a result, Rabin's algorithm can be reliable even if only one number is chosen at random.

Unfortunately, there are a couple of reasons why Bach's result does not solve all of our problems. First, results of this sort (Bach [1991]; Karloff and Raghavan [1993]) are only available for a few randomized algorithms. Second, while Bach's result shows that there are very few bad sequences in the small subset of sequences that an LCG is capable of producing, it does not establish that bad sequences are unlikely to be chosen—and that Rabin's algorithm is thus reliable—unless a_1 is chosen at random. In practice, the value of a_1 might be chosen in such a way that the bias is toward the few bad sequences in the small subset.

There are essentially two ways to eliminate the worry that the value of a_1 might be chosen in such a way that the bias is toward bad sequences. First, we can try to eliminate the need to choose even one number at random. As the computer scientist Richard Karp points out, in some cases, 'at the cost of some additional computation, randomness can be eliminated entirely' ([1991],

p. 198). That is, developing a randomized algorithm can sometimes serve as a heuristic device for developing a deterministic algorithm for the same task. In fact, Rabin's algorithm can be turned into a deterministic algorithm that is fast and that is provably reliable—on the assumption that the Extended Riemann Hypothesis is true. Unfortunately, a deterministic algorithm that is almost as fast—as its randomized counterpart—and that is provably reliable is not always readily available. In fact, that is why computer scientists have resorted to randomized algorithms in the first place.

Second, we can try to find a way to pick the value of a_1 that is unlikely to lead to bad sequences—e.g. we can try to find a way to pick the value of a_1 at random. The value of a_1 is typically chosen by sampling the system clock when the randomized algorithm is executed. There is no particular reason to believe that sampling the system clock is any more likely to lead to bad sequences than is a truly random process. However, since it is not clear that there is any scientific or mathematical evidence that sampling the system clock is unlikely to lead to bad sequences, mathematicians may still have reason to suspend judgement on the truth of mathematical propositions that Rabin's algorithm has been used to establish. In other words, the mere fact that we have no evidence that a claim is false is not always good grounds for believing that it is true.[9] As a result, we may have to fall back on our second option—for dealing with the false premise in the proofs of reliability—and try to find a way to pick numbers at random.

5 Physical sources of randomness

The second option—for dealing with the false premise in the proofs of reliability—is to find a way for randomized algorithms to pick numbers at random. Fortunately, there seem to be several ways to pick numbers at random. For example, it is often asserted, as it has been by Ivar Ekeland, that 'in quantum mechanics, to measure means to draw at random' ([1993], p. 31). In particular, we might generate random bits by measuring the Y spin of X spin up particles (see e.g. Albert [1992], pp. 1–16). Unfortunately, there are at least three difficulties with using quantum phenomena—or any other physical source of randomness—to pick numbers at random. The first two difficulties have been substantially taken care of by recent technical developments. However, the third difficulty is somewhat less tractable.[10]

[9] According to Douglas Walton ([1992]), such *arguments from ignorance* are not always fallacious in everyday argumentation. It is not clear, however, if such arguments are sufficiently compelling to support mathematical or scientific conclusions.

[10] The probabilistic method discussed in Fallis [1997] uses a physical source of randomness instead of an RNG. However, since even physical sources of randomness are suspect, there is almost as much reason to suspend judgement on the reliability of that method as there is to suspend judgement on the reliability of randomized algorithms.

First, it tends to be 'slow and costly to generate' (Karp [1991], p. 196) random numbers with a physical source of randomness. The fact that physical sources tend to be slow is especially annoying since the main motivation for using a randomized algorithm is the fast execution time. However, recent developments may have solved this particular difficulty. For instance, a CD-ROM is now available which contains about five billion bits derived from a physical source of randomness (see Peterson [1998], pp. 178–80). Also, there is now an inexpensive device which can be plugged into the parallel port of a personal computer and which generates thousands of bits per second from a physical source of randomness (see Quantum World Corp. [1995]).

Second, as Miklos Santha and Umesh Vazirani point out, 'the available physical sources of randomness—including zener diodes and geiger counters—are imperfect. Their output bits are not only biased but also correlated' ([1986], p. 75). In other words, physical sources of randomness do not have an equal chance of choosing any particular sequence of numbers from a set of sequences. Again, recent developments may have solved this particular difficulty. Computer scientists have developed algorithms which 'convert the output of such sources into bit-sequences that are provably good for computational applications' such as 'primality testing' (*ibid.*, p. 75). The only requirement on the input to these algorithms is that there is an upper bound on the probability that the next bit is 1 rather than 0 and that there is a lower bound on this probability. In other words, the physical source just has to be a 'slightly random source' (*ibid.*, p. 76).

Third, it is not clear that there is compelling scientific evidence for the claim that physical sources are even 'slightly random'. For example, one common physical source of randomness that is not even 'slightly random' is a coin flip. The outcome of a coin flip is completely determined by the initial conditions. As a result, the probability that the outcome will be heads is either 1 or 0.[11] Of course, we do not usually know enough about the initial conditions in order to predict the outcome of a coin flip. However, while randomness implies unpredictability, the converse does not hold. In short, the fact that a process is unpredictable does not imply that it has an equal chance of choosing any particular sequence from a set of sequences.[12]

Interestingly enough, the outcome of a measurement of some quantum phenomena may also be completely determined by initial conditions. According to most of the competing theories of quantum mechanics, the outcome of such a measurement is a random event. However, there are descriptively

[11] It might be suggested that a *series* of coin flips would produce a random sequence of heads and tails because the initial conditions would vary randomly from flip to flip. However, since the initial conditions for each flip are determined by the initial conditions for the series of coin flips, the probability of any particular sequence of heads and tails being the outcome is either 1 or 0.

[12] A common use of randomness is in the encoding of secret messages (see e.g. Peterson [1998], pp. 124–6). In this case, the unpredictability of the source of randomness *is* the crucial concern.

adequate theories of quantum mechanics that are deterministic. For example, according to David Albert, 'the account which Bohm's theory gives of [the motions of particles] is *completely deterministic* [. . .] and so the very *idea* of probability will have to enter into this theory as some kind of an *epistemic* idea, just as it enters into classical statistical mechanics' ([1992], p. 135).[13] In other words, quantum phenomena might just be obeying an extremely complex deterministic algorithm that we are not able to distinguish from a truly random process. As a result, there is some reason to suspend judgement on the randomness of quantum phenomena.

Fortunately, whether or not Rabin's algorithm chooses numbers at random is not what is crucial in order for a proof of reliability to go through. As we have seen above, all that needs to be shown is that bad sequences are unlikely to be chosen. Thus, in order to establish the reliability of Rabin's algorithm implemented with a physical source of randomness, we would just need to show that the physical source of randomness in question is unlikely to produce bad sequences.

6 'No experimental data are cited to support this claim'

There is certainly no reason to believe that quantum phenomena—or any other physical source of randomness—are likely to produce bad sequences for Rabin's algorithm. Quantum mechanics, one of our best confirmed scientific theories, provides no apparent mechanism for a correlation between quantum phenomena and bad sequences for Rabin's algorithm. Even so, no one has performed an experiment to establish that quantum phenomena are unlikely to produce bad sequences.

According to the philosopher George Berkeley, 'it is to me a sufficient reason not to believe the existence of anything, if I see no reason for believing it' ([1979], p. 53). However, this is a somewhat questionable principle to use in order to establish that there is no correlation between quantum phenomena and bad sequences for Rabin's algorithm. After all, there have been many cases where scientists have discovered an unexpected correlation between physical phenomena and mathematical formulas. For example, the number of petals on flowers—and the number of spirals on the face of a giant sunflower—is, rather unexpectedly, always a number in the Fibonacci sequence (see e.g. Stewart [1995], pp. 135–6). Thus, in the absence of direct empirical evidence that there is no correlation between quantum phenomena and bad

[13] I should make a couple of remarks about Bohm's theory here. First, even though Bohm's theory is a hidden variable theory, it has not been ruled out by those experiments that show that *local* hidden variable theories will not work (*cf.* Albert [1992], pp. 68–9, 158–9). Second, unlike a coin flip, we cannot predict the outcome of a measurement of quantum phenomena even in principle according to Bohm's theory. However, as I noted above, unpredictability is not the important issue for our purposes.

sequences, mathematicians might still be inclined to suspend judgement on the reliability of Rabin's algorithm—and on the truth of any mathematical propositions that it has been used to establish.

Rather than try to provide compelling evidence for the claim that quantum phenomena are unlikely to produce bad sequences, I will try to achieve a more modest goal. Using an example developed by Peter Urbach ([1985], pp. 264–5), I will argue that, *in order to do science at all*, scientists have to accept many such claims even though 'no experimental data are cited to support [these] claim[s]' (*ibid.*, p. 264). As a result, it would be inconsistent for scientists to suspend judgement on the reliability of Rabin's algorithm on the grounds that there is no direct experimental evidence that quantum phenomena are unlikely to produce bad sequences.

Suppose that a scientist wants to establish that the consumption of alcohol slows down reaction times. Scientists typically confirm such hypotheses by running a *controlled experiment*. The scientist assigns subjects to either an *experimental group* or to a *control group*. She gives alcohol to the experimental group and she gives no alcohol to the control group. After testing the reaction times of both groups, the scientist would like to be able to conclude that any disparity between the reaction times of the two groups is due to the consumption of alcohol. However, she cannot draw this conclusion if there is some unintended difference between the two groups—e.g. one group is significantly older than the other—which could possibly have an effect on the outcome of the experiment. As a result, she has to insure that the only difference between the two groups that could possibly have an effect on the outcome of the experiment is the consumption of alcohol.

One way to insure that there are no unintended differences between the two groups is to *control for* all of those factors that could possibly have an effect on the outcome of the experiment. For example, the scientist might assign subjects to the experimental and control groups so that the average age of the two groups is roughly the same. However, scientists do not control for everything. In practice, scientists only control for those things that, according to the prevailing scientific theories, are likely to have an effect on the outcome of an experiment. As a result, this scientist is probably not going to control for 'the color of the subjects' eyes [. . .] since this is ''almost certainly negligible'' as an influence on their reaction times' (*ibid.*, p. 264). In doing so, the scientist is assuming that eye color is unlikely to have an effect on the outcome of this experiment even though 'no experimental data are cited to support this claim'.

Not only does the scientist actually make such assumptions, but she has to make such assumptions. First, there is an unending list of factors—some of which may not even have occurred to the scientist—that *could* possibly have an effect on the outcome of the experiment. The scientist cannot explicitly

control for all of them. Second, while the scientist might try to avoid making any particular assumption—e.g. by performing an experiment to establish that eye color has no influence on reaction time, this strategy would lead to an infinite regress. As Peter Urbach points out, 'even if a careful trial had been made [to establish that eye color has no influence on reaction time], certain conceivable influences on *its* outcome would have to have been set aside as negligible' (*ibid.*, p. 264). This sort of regress would occur in the physical sciences as well as the human sciences.

Nevertheless, there is a way for scientists to control for everything that could possibly have an effect on the outcome of an experiment in one fell swoop. Namely, scientists can assign subjects to the experimental and control groups *at random*—and, in fact, this is what scientists commonly try to do. *Randomization* does not guarantee that there will be no differences—other than the consumption of alcohol—that could possibly have an effect on the outcome of this experiment. However, if a truly random process is used to assign subjects to the experimental and control groups, then it is *unlikely* that there will be such a difference between the two groups. In other words, just as we found with Rabin's algorithm, a truly random process is unlikely to produce a 'bad sequence'.

Unfortunately, as we have seen, it is not clear that anybody has access to a truly random process. Instead, scientists have to use 'the throw of a standard coin or die, the draw of a card from a well-shuffled pack, or the decay of a radioactive element' (*ibid.*, p. 258) to assign subjects to the experimental and control groups. However, just as no one has ever performed an experiment to establish that quantum phenomena are unlikely to produce bad sequences for Rabin's algorithm, no one has ever performed an experiment to establish that quantum phenomena are unlikely to produce bad sequences for reaction time experiments. As a result, a scientist has to assume that her source of randomness is unlikely to lead to unintended differences between the experimental and control groups even though 'no experimental data are cited to support this claim'.

Since scientists have to assume, for example, that quantum phenomena are unlikely to produce bad sequences for reaction time experiments, even though 'no experimental data are cited to support this claim,' they should not shrink from assuming that quantum phenomena are unlikely to produce bad sequences for Rabin's algorithm. In fact, they also should be willing to assume that flipping a coin—or sampling the system clock to choose the value of a_1—is unlikely to produce bad sequences.

These considerations suggest that it would be inconsistent for *scientists* to suspend judgement on the reliability of Rabin's algorithm implemented with a physical source of randomness. However, it is not clear that it would be inconsistent for *mathematicians* to suspend judgement on the reliability of

Rabin's algorithm. After all, mathematicians do not perform controlled experiments in order to establish that mathematical propositions are true. In point of fact, mathematics can be—and has been—carried out quite successfully without any appeal to empirical facts.

There are important cases, however, where mathematicians have made use of empirical facts in order to establish that mathematical propositions are true. For example, part of the current justification for the truth of the four-color theorem is the fact that the electronic components of a digital computer behave in certain predictable ways (see e.g. Teller [1980] and Deltefsen and Lukar [1980]). In the absence of any rationale for accepting certain forms of scientific justification and rejecting others, it does not seem to be consistent for mathematicians to suspend judgement on the reliability of Rabin's algorithm, but not suspend judgement on the truth of the four-color theorem.

For the sake of completeness, I should note that there is always some reason for an individual to suspend judgement on the truth of mathematical propositions that a randomized algorithm has been used to establish. After all, it is possible that 'some malicious demon of the utmost power and cunning has employed all his energies in order to deceive me' (Descartes [1988], p. 79). However, if mathematicians were to suspend judgement on the reliability of randomized algorithms on the basis of *demon doubt*, then they would have to suspend judgement on much more than just the truth of the four-color theorem. For example, a mathematician would then have to worry about going 'wrong every time I add two and three or count the sides of a square, or in some even simpler matter, if that is imaginable' (*ibid.*, p. 79).

7 Concluding remarks

In the preceding section, I argued that it would be inconsistent for mathematicians to suspend judgement on the reliability of Rabin's algorithm when a physical source of randomness is used to choose the sequence of numbers. In addition, this implies—given Bach's result—that they should not suspend judgement when Rabin's algorithm is implemented with an LCG and a physical source of randomness is used to choose the value of a_1. As a result, there are conditions where nothing should stop mathematicians from using probabilistic methods to establish that mathematical propositions are true. Unfortunately, it is not clear that such conditions obtain very often. In most cases, randomized algorithms are implemented with RNGs even though it has not been *proven*—à la Bach—that there are very few bad sequences in the small subset of sequences that the RNGs are capable of producing.

Unlike the question of whether or not there is a correlation between quantum phenomena and bad sequences for a particular randomized algorithm, the question of whether or not there is a correlation between a particular RNG

and bad sequences for a particular randomized algorithm is purely mathematical. Furthermore, it is not clear that correlations between two such mathematical formulas are entirely unexpected. For example, prior to Bach's result, a correlation between an LCG and bad sequences for Rabin's algorithm would not have been terribly unexpected. As we saw above, an LCG uses large primes and modular arithmetic to produce a sequence of numbers. At the same time, the definition of a witness in Rabin's algorithm involves modular arithmetic and is used to identify large primes (see Rabin [1980], p. 130). Even so, this does not necessarily imply that mathematicians should only accept that a particular randomized algorithm implemented with a particular RNG is reliable when they have a proof of this fact. After all, computer scientists have pretty good evidence for the—purely mathematical—claim that **P** is not equal to **NP** even though they do not have a proof (see e.g. Harel [1989], p. 162).

Finally, I should note that there is still a small problem even when Rabin's algorithm is implemented with a physical source of randomness. Namely, there is the problem of convincing *other* mathematicians that the sequence of numbers used by the algorithm really was generated by a physical source of randomness. While mathematicians can tell just by looking that a deductive proof is valid, they cannot tell just by looking that a sequence of numbers has been generated by a physical source of randomness. This is because *any* particular sequence of numbers might have been generated by a random process. Even so, I would imagine that mathematicians could use the same sorts of techniques that scientists already use to assure themselves that another scientist has not faked her data.

Acknowledgements

I would like to thank Jeff Barrett, Martin Frické, Charlie Hurt, Peter Lewis, Pen Maddy, Kay Mathiesen, Catherine Womack, Peter Woodruff, and Dan Zelinski for reading and commenting on this work at various stages in its development.

School of Information Resources
University of Arizona
fallis@u.arizona.edu

References

Albert, D. Z. [1992]: *Quantum Mechanics and Experience*, Cambridge, MA: Harvard University Press.

Bach, E. [1991]: 'Realistic Analysis of Some Randomized Algorithms', *Journal of Computer and System Sciences*, **42**, pp. 30–53.

Berkeley, G. [1979]: *Three Dialogues between Hylas and Philonous*, in R. M. Adams (*ed.*), Indianapolis: Hackett.

Descartes, R. [1988]: *Selected Philosophical Writings*, J. Cottingham, R. Stoothoff, and D. Murdoch (*trans.*), New York: Cambridge University Press.

Detlefsen, M. and Lukar, M. [1980]: 'The Four-Color Theorem and Mathematical Proof', *Journal of Philosophy*, **76**, pp. 803–20.

Ekeland, I. [1993]: *The Broken Dice*, Chicago: University of Chicago Press.

Fallis, D. [1997]: 'The Epistemic Status of Probabilistic Proof', *Journal of Philosophy*, **94**, pp. 165–86.

Ferrenberg, A. M., Landau, D. P., and Wong, Y. J. [1992]: 'Monte Carlo Simulations: Hidden Errors From ''Good'' Random Number Generators', *Physical Review Letters*, **69**, pp. 3382–4.

Harel, D. [1989]: *The Science of Computing*, Reading, MA: Addison-Wesley.

Hersh, R. [1997]: *What Is Mathematics, Really?*, New York: Oxford University Press.

Karloff, H. J. and Raghavan, P. [1993]: 'Randomized Algorithms and Pseudorandom Numbers', *Journal of the Association for Computing Machinery*, **40**, pp. 454–76.

Karp, R. M. [1991]: 'An Introduction to Randomized Algorithms', *Discrete Applied Mathematics*, **34**, pp. 165–201.

Knuth, D. E. [1981]: *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, second edn., Reading, MA: Addison-Wesley.

Levi, I. [1984]: 'Conflict and Social Agency', in *Decisions and Revisions: Philosophical Essays on Knowledge and Value*, Cambridge: Cambridge University Press, pp. 257–70.

Motwani, R. and Raghavan, P. [1995]: *Randomized Algorithms*, New York: Cambridge University Press.

Park, S. K. and Miller, K. W. [1988]: 'Random Number Generators: Good Ones Are Hard to Find', *Communications of the Association for Computing Machinery*, **31**, pp. 1192–201.

Peterson, I. [1998]: *The Jungles of Randomness: A Mathematical Safari*, New York: John Wiley & Sons.

Pomerance, C. [1981]: 'Recent Developments in Primality Testing', *Mathematical Intelligencer*, **3**, pp. 97–105.

Quantum World Corp. [1995]: 'What is the QNG Device?', http://comscire.com/QNGINFO.html (visited 25 July 1999).

Rabin, M. O. [1980]: 'Probabilistic Algorithm for Testing Primality', *Journal of Number Theory*, **12**, pp. 128–38.

Ripley, B. D. [1988]: 'Uses and Abuses of Statistical Simulation', *Mathematical Programming*, **42**, pp. 53–68.

Santha, M. and Vazirani, U. V. [1986]: 'Generating Quasi-Random Sequences From Semi-Random Sources', *Journal of Computer and System Sciences*, **33**, pp. 75–87.

Sinisalo, M. K. [1993]: 'Checking the Goldbach Conjecture Up to 4.10^{11}', *Mathematics of Computation*, **61**, pp. 931–4.

Skyrms, B. [1996]: *Evolution of the Social Contract*, New York: Cambridge University Press.

Solovay, R. and Strassen, V. [1977]: 'A Fast Monte-Carlo Test for Primality', *SIAM Journal on Computing*, **6**, pp. 84–5.

Stewart, I. [1995]: *Nature's Numbers*, New York: Basic Books.

Teller, P. [1980]: 'Computer Proof', *Journal of Philosophy*, **76**, pp. 797–803.

Urbach, P. [1985]: 'Randomization and the Design of Experiments', *Philosophy of Science*, **52**, pp. 256–73.

Walton, D. [1992]: 'Nonfallacious Arguments From Ignorance', *American Philosophical Quarterly*, **29**, pp. 381–7.

Womack, C. and Farach, M. [1997]: 'Randomization, Rigor and Persuasiveness in Proofs', paper presented at the *International Meeting on Logic and Mathematical Reasoning*, Mexico City.

Zeilberger, D. [1993]: 'Theorems for a Price: Tomorrow's Semi-Rigorous Mathematical Culture', *Notices of the American Mathematical Society*, **40**, pp. 978–81.