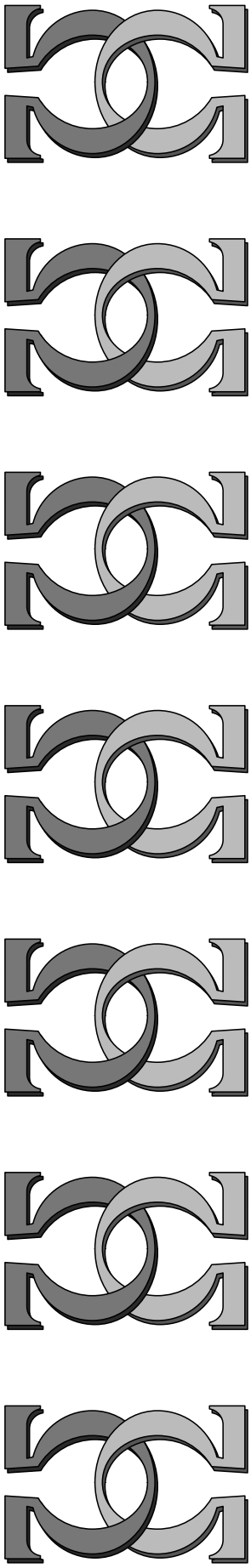

CDMTCS
Research
Report
Series

Experimental Evidence of the Incomputability of Quantum Randomness

A. A. Abbott[1], C. S. Calude[2], M. J. Dinneen[2] and N. Huang[2]

[1] University Grenoble Alpes, CNRS, France
[2] University of Auckland, New Zealand

CDMTCS-515
November 2017

Centre for Discrete Mathematics and
Theoretical Computer Science

Experimental Evidence of the Incomputability of Quantum Randomness

Alastair A. Abbott[*], Cristian S. Calude[†], Michael J. Dinneen[†] and Nan Huang[†]

November 26, 2017

Abstract

To be written.

1 Introduction

Randomness is an important resource in a diverse range of domains: it has uses in science, statistics, cryptography, gambling, and even in art and politics. In many of these domains, it is crucial that the randomness be of high quality. This is most directly the case in cryptography, where good randomness is vital to the security of data and communication, but is equally, albeit more subtly, true in other areas such as politics, where decisions of consequence may be made based on scientific and statistical studies relying crucially on randomness.

For a long time, people have predominantly relied on pseudo random number generators (PRNGs) – that is, computer algorithms designed to simulate randomness – to serve such needs. Problems with various PRNGs, often only uncovered when it is already too late, are all too common and can have serious consequences.[1] This has driven a recent surge of interest in RNGs exploiting physical phenomena, and more particularly in quantum RNGs (QRNGs) that utilize the purportedly inherent randomness in quantum mechanics [16, 48, 55, 57]. QRNGs are generally considered to be, by their very nature, better than classical RNGs (such as PRNGs), but how (or can) one test this in practice?

RNGs are generally tested by conducting batteries of tests on (finite) sequences they have produced [41, 49]. Traditionally, such tests have focused on intuitive aspects of randomness, such as the frequencies of certain (strings of) bits, but human intuition about randomness is notoriously poor [14, 31] and many other symptoms of randomness remain untested. Indeed, the randomness of strings and sequences is an incomputable property and thus cannot be verified completely and, moreover, it is characterised by an infinity of properties [22]. With standard randomness tests designed with PRNGs in mind, it is reasonable to wonder whether there are tests more appropriate for analysing QRNGs and perceiving the advantage they can provide. Indeed, QRNGs should excel precisely on properties of randomness where algorithmic PRNGs are doomed to fail: incomputability and their inherent unpredictability [5, 7, 9, 24]. Although incomputability is not directly testable (not least because it is a property of infinite sequences and thus holds only in the limit), one may ask whether there are tests than can reveal related advantages in practice.

[*]University Grenoble Alpes, CNRS, Grenoble INP, Institut Néel, 38000 Grenoble, France

[†]Department of Computer Science, University of Auckland, Private Bag 92019, Auckland, New Zealand.

[1]An example is the discovery in 2012 of a weakness in the encryption system used worldwide for online shopping, banking, email; the flaw was traced to the numbers a PRNG has produced [39]. As of 2017, Java still relies on a linear congruential generator, a low quality PRNG.

With this goal, we study several possible tests of randomness based on algorithmic information theory. In particular, we consider tests based on Borel normality [18, 19] as well as novel tests based on the Solovay-Strassen probabilistic primality test [54] – an algorithm which can be made deterministic when given access to algorithmic randomness [26]. We test several classical RNGs as well as a semiconductor-based QRNG [38] with these tests and observe a clear advantage of the QRNG in the latter test.

2 Randomness

In order to guide the development of tests for QRNG, it is important to understand what randomness is and thus what one should test. Historically, the quest to develop a formal understanding of randomness focused on the problem of determining whether a given (finite) string or (infinite) sequence of bits is random. One of the first attempts to formalise such a notion of randomness is due to Borel, who defined the concept of *Borel normality* for infinite sequences [18]. Borel normality formalises the notion that bits should be evenly and equally distributed within a sequence. Although this captures one of the most intuitive features of randomness, it does not alone capture fully the desired concept. For example, the *Champernowne sequence* $0\,1\,00\,01\,10\,11\,000\,001\,011\,100\ldots$ [27] contains every string of length k with the same limiting frequency of 2^{-k}, and yet the sequence has a simple description: concatenate the binary representation of all the strings of length k in lexicographical order for $k = 1, 2, \ldots$. Given this description, it is clear that the Champernowne sequence is not random, but highly ordered.

The study of algorithmic information theory, developed in the 1960s by Solomonoff, Kolmogorov and Chaitin, provides a more robust and acceptable definition of random sequence. In this framework, random sequences are those that are asymptotically incompressible [25] or, equivalently, pass all effective Martin-Löf tests [44]. Incompressibility also allows notions of randomness for finite strings to be formalised, although the notion is only defined up to a choice of universal Turing machine, so that the notions only converge in the limit of long strings [22].

While algorithmic information theory thus provides a sound notion of randomness for strings and sequences, two important points must be mentioned. Firstly, it is not effectively decidable whether a string or sequence is random, so the notion does not provide an effective way to test the randomness of a sequence of bits. Secondly, it is possible to define ever stronger notions of randomness: from an algorithmic perspective, no notion of "pure" or "absolute" randomness exists, only degrees of randomness [22, 23, 33]. *This should temper any desire to verify the randomness of a RNG by tests on its output. Instead, we can only hope to compare the quality of strings produced.*

As interest in *generating* random numbers increased, the concept of randomness received increased philosophical attention and it became clearer that the algorithmic notion of randomness fails to capture aspects of randomness important for RNGs [3]. Indeed, as von Neumann noted, "there is no such thing as a random number – there are only methods to produce random numbers" [58]. The insight of von Neumann is not that the algorithmic notion of randomness is problematic – indeed, it is highly satisfactory as a notion of random *objects* – but that there is a dual concept of randomness, that of random *processes* [3, 30]. Such a concept has historically received little attention, but the most convincing attempts to make it rigorous are perhaps those which define it as a form of maximal unpredictability: the outcome of such a process should be unpredictable for any physical observer [8, 29].

There are thus two legitimate notions of randomness to be reconciled: that of process randomness, formalised by the uniform probability measure; and that of product randomness, formalised by algorithmic randomness. The distinction between these notions is important for understanding

tests of randomness.

3 Random number generators

An ideal random number generator is generally taken to be a random process producing the same probability distribution as the ideal (but unphysical) unbiased coin. It thus produces bits sequentially generating a sequence $\mathbf{x} = x_1 x_2 \ldots$ with each bit x_i being equiprobable, i.e. $p(x_i = 0) = p(x_i = 1) = 1/2$, and with successive bits produced independently. Thus, all strings x of length k have probably $p(x) = 2^{-k}$ and, in the infinite limit, one obtains the Lebesgue measure over all infinite sequences [21].

If one tries to implement such a device in practice, two issues immediately become apparent.

Firstly, how is one to know that the process exploited is really random and actually produces the expected ideal distribution? This issue touches on the interpretation of probability [34] (although this is beyond the scope of the present article). For example, a physical process thought to be represented by the uniform distribution might only exhibit epistemic randomness, and a more precise, deterministic model of the process might be possible which reveals its non-randomness. The most direct way to avoid such possibilities is to harness an indeterministic process to ensure its unpredictability [8].

Secondly, how does one test or verify the randomness of a RNG given that one only has access to (finite) strings produced by it? Although the concepts of process and product randomness are indeed distinct, they are nonetheless related: long enough strings produced by an ideal RNG will, with high probability, be random, while in the infinite limit the sequences produced will be algorithmically random with probability 1 *but not certainty*: an ideal coin can in principle produce non-random or even computable sequences. However, as mentioned earlier, the randomness of sequences is already an incomputable property. Thus, one can do no better than verifying finitely many properties of randomness to gain confidence in a RNG.

3.1 Pseudo RNGs (PRNGs)

The predominant approach to generating randomness is to use algorithms to produce "pseudo-randomness", and such PRNGs are ubiquitous as a result of their practicality and speed. However, the very fact that such devices use computational methods to produce their outcomes distinguishes them from ideal RNGs. PRNGs typically use a short string from an external source – generally assumed to be random – as an initial "seed" for an algorithm [2]. Thus, PRNGs can only produce computable sequences, whereas such sequences should be produced only with probability 0 by an ideal RNG. Instead, effort is made to make PRNGs difficult to distinguish from an ideal RNG given limited (typically polynomial time) computational resources [32], since this provides a degree of security against cryptographic attacks, even if the resulting distribution (induced by the distribution over the initial seeds) is far from uniform in reality.

PRNGs generally produce sequences that satisfy many intuitive aspects of randomness – such as the equidistribution of the bits produced – and pass most standard statistical tests of randomness despite their computability. Nonetheless, deficiencies resulting from the non-randomness of PRNGs are regularly exploited (e.g. [17]) and much of the interest in quantum randomness has been driven by the potential to avoid the shortcomings of PRNGs.

4 Quantum randomness

For some time now, quantum mechanics has garnered interest as a potential source of randomness for RNGs. Such interest stems from the fact that certain quantum phenomena, such as the radioactive decay of an atom or the detection of a photon having passed through a beamsplitter, are generally taken to be "intrinsically random" under the standard interpretation of quantum mechanics [11]. We will first discuss these claims in a little more detail – since it is important to base the randomness of QRNGs on more formal grounds rather than simply assuming such randomness – before discussing one approach to the generation of quantum randomness in more detail.

Claims about quantum randomness originate with the fact that, as a formal theory, quantum mechanics differs fundamentally from classical physics in that not all observable properties are simultaneous defined with arbitrary precision. Instead, quantum mechanics, via the Born rule, only specifies the probabilities with which individual measurement outcomes occur for the measurement of a physical quantity – i.e., a quantum *observable*. Formally, if a system is in a quantum state $|\psi\rangle$ and one measures an observable A with spectral decomposition $A = \sum_i a_i P_i$, where we adopt the notation $P_i = |i\rangle\langle i|$ for rank-1 projection observables, then one obtains outcome a_i with probability

$$P(a_i|\psi) = |\langle i|\psi\rangle|^2. \tag{1}$$

Thus, whereas randomness in classical physics is due to ignorance of the precise initial conditions of a systems (e.g., as in chaotic systems) [40], in quantum mechanics it is intrinsic to the formal theory.

Nonetheless, the Born rule is a purely formal statement, and interpreting the probability distribution specified by the Born rule remains the subject of ongoing debate. The orthodox interpretation, however, is that the distribution should be understood ontically as representing an indeterministic phenomenon [11]. Crucially, this interpretation is more than a blind assumption: several well-known no-go theorems rule out classical statistical interpretations of quantum randomness.

Bell's Theorem [15] is the most well-known of these results, and shows that a classical, local hidden variable theory cannot reproduce the statistics of quantum correlations that are observed [12] between entangled particles. The Kochen-Specker Theorem [37], although perhaps lesser known, pinpoints this breakdown in determinism in an arguably more precise way: it shows that, for any quantum system with more than 2 dimensions, it is logically impossible to predetermine all measurement outcomes prior to measurement in a noncontextual fashion (i.e., in a way which is independent of other compatible – and thus non-disturbing – measurements one can perform).

More recently, this theorem has been refined to show that the only observables that can be predetermined in a noncontextual way are those for which the Born rule assigns the probability 1 to a particular outcome [6, 10]. More precisely, we call an observable A *value definite* for a state $|\psi\rangle$ if it has a predetermined measurement outcome $v_\psi(A)$. This stronger result shows that for systems of more than 2 dimensions, if we assume that any such value definite observables should be noncontextual, then A is value definite if and only if $|\psi\rangle$ is an eigenstate of A; all other observables must be *value indefinite*.

This result makes the extent of quantum value indefiniteness – and thus indeterminism – clear and pinpoints which measurements are protected by such formal results, allowing the randomness of QRNGs to be based more rigorously on physical principles and clarifying its link to indeterminism. Crucially, this result also allows one to show that the measurement of such value indefinite observables satisfies a strong form of unpredictability [9], proving that one really can-

not provide better predictions than the Born rule specifies, and thus giving a stronger theoretical grounding to claims about the form of quantum randomness proposed for QRNGs.

4.1 Quantum RNGs (QRNGs)

These properties of quantum measurements make it an ideal candidate for random number generation: if one measures an observable for which the Born rule predicts a uniform distribution, then the QRNG embodies a perfect coin. Moreover, the results discussed above show that – subject to very reasonable physical assumptions about how classical objects should behave – this distribution can't be given an epistemic interpretation and is truly of indeterministic origin. The attractiveness of QRNGs is further enhanced by the possibility of obtaining high bitrates and the simplicity of their physical models – in contrast to RNGs based on classical physics, such as chaotic systems.

Early QRNGs relied on features such as radioactive decay [51], but simpler systems based, for example, on measuring the polarisation [36, 52, 55] or detection time [56] of a photon, have become the norm due to the practical advantages they provide. Such approaches have led to the development of commercial QRNGs, such as ID Quantique's Quantis [35].

Many successful QRNGs exploit 2-dimensional systems to generate randomness (e.g. Quantis uses the polarisation of photons). This greatly simplifies the design and production of such devices but neither Bell's theorem (which requires entanglement) nor the Kochen-Specker Theorem (which requires at least 3-dimensional systems) are applicable, and these QRNGs thus lack the rigorous theoretical certification that quantum mechanics can provide, even if it is reasonable to think that the measurements they exploit should still be indeterministic.

More recently there has been significant interest in implementing QRNGs that violate Bell's inequalities in order to provide a much stronger certification [28, 48]. Specifically, such devices allow the indeterminism of a QRNG to be certified in a device-independent way – i.e., without assuming knowledge of how the device works – which is particularly important in cryptographic settings, where the workings of a RNG should perhaps not be trusted. Such certification, however, comes at a cost, since not only does it require an initial small random seed anyway, but it also relies on the QRNG being separated into two space-like separated (or at least isolated) components and the stringent requirements of loophole-free Bell tests reduce the obtainable bitrate by several orders of magnitude compared to "standard" QRNGs [48].

An alternative approach outlined in [5, 7] is to use 3-dimensional systems exhibiting value indefiniteness (via the Kochen-Specker Theorem) to certify a QRNG. While such a certification is device dependent (i.e., one relies on knowledge of the functioning of the QRNG), it allows the practical advantages of standard QRNGs to be maintained while providing stronger theoretical certification. Figure 1 shows the schematic of the proposed QRNG, which uses a spin-1 particle as a 3-dimensional system. The system is prepared in the state $|S_z = 0\rangle$ (i.e., with spin-0 in the z-direction) before measuring the spin in the x-direction, i.e. the observable $S_x = \sum_{s_x=-1}^{1} s_x |S_x = s_x\rangle\langle S_x = s_x|$. Since the state $|S_z = 0\rangle$ is an eigenstate of the projection observable $P_0 = |S_x = 0\rangle\langle S_x = 0|$, this observable is value definite with value $v(P_0) = 0$ – that is, the result of the S_x measurement is never $S_x = 0$. However, by the results of [5, 10], both $P_\pm = |S_x = \pm 1\rangle\langle S_x = \pm 1|$ are value indefinite and, moreover, both outcomes $S_x = \pm 1$ occur with probability $1/2$ according to the Born rule (1). Thus, the QRNG operates as an ideal coin certified by value indefiniteness.

This approach to certifying a QRNG via value indefiniteness leads to some interesting additional consequences if one is willing to accept slightly stronger physical assumptions (in particular, about whether being able to compute properties in advance implies well-defined physical prop-

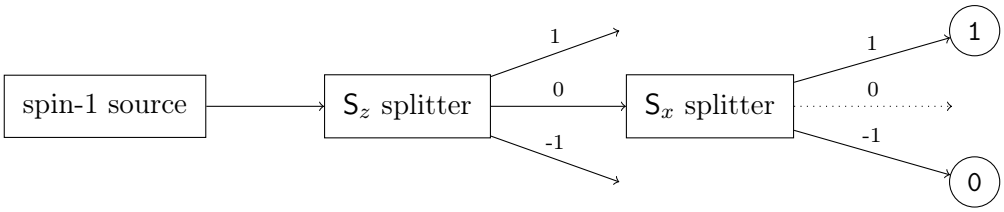

Figure 1: Schematic of a QRNG certified by value indefiniteness exploiting the measurement of a spin-1 particle [5]. *[It doesn't make any sense to give the generalised beamsplitter setup too because it is actually less related to what Arkady et al. implemented.]*

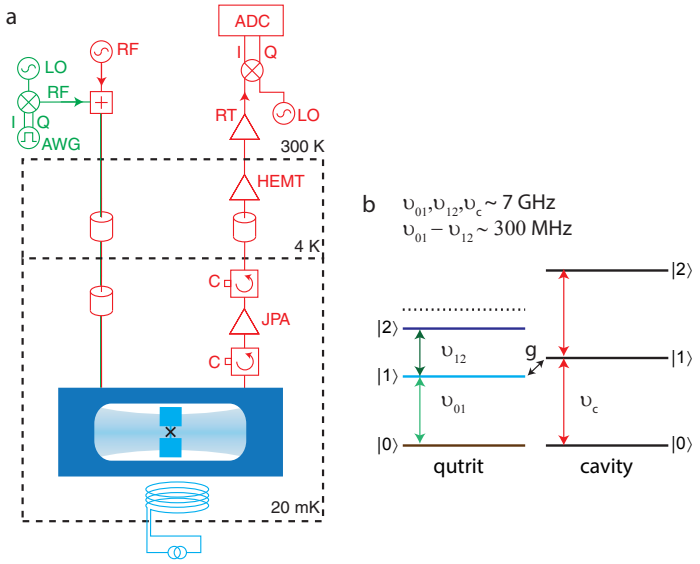

Figure 2: An implementation of a qutrit as a superconducting transmon coupled to a microwave cavity [38]. *[I'm not convinced we should really include this figure since without further technical details it doesn't really mean much.]*

erties). Specifically, it was shown in [5] that such a device, if used repeatedly *ad infinitum* to generate an infinite sequence $\mathbf{x}$ of bits, will produce an $\mathbf{x}$ that is strongly incomputable (technically, "bi-immune") not just with probability 1 but *with certainty*. Although such a result will not alone lead to observable advantages for finite strings – recall that an ideal QRNG will produce an incomputable sequence with probability 1 – but this nonetheless highlights the differences between pseudo and quantum randomness in relation to computability.

The QRNG outlined in Fig. 1 has recently been implemented experimentally [38], not with spin-1 particles but by exploiting a superconducting transmon coupled to a microwave cavity as a qutrit, a system isomorphic to the spin-1 particle described above; see Fig. 2. The authors used this QRNG implementation to generate a large number of bits, and in the subsequent sections we will analyse these sequences produced from quantum randomness. In particular, we will focus on observing differences between such sequences and pseudorandom ones arising from algorithmic properties of the sequences.

5 Testing RNGs

While it is crucial to have a good theoretical understanding of any RNG, there are several reasons why testing experimentally their performance is nonetheless crucial. Firstly, one can never be sure that the implementation of the RNG matches the theory, a fact that is equally as true for hardware RNGs as for software RNGs. Indeed, in the extreme limit, one might not wish to trust any theoretical claims about about a given RNG, and thus confidence in the RNG can only be gained from performing carefully selected tests. Secondly, thorough testing gives one the opportunity to detect any issues with assumptions made in the theoretical analysis of a device or in its practical deployment (e.g., if the distribution of seeds does not match that assumed theoretically the performance of a RNG might be compromised).

It is nonetheless important to recognise that experimental testing can never allow one to perfectly characterise a device. Instead, with access only to finite strings produced by it and the ability to perform a finite number of tests, one can only ever gain increasing confidence in the operation of the device. One can never be sure, for instance, that the output obtained was not simply atypical behaviour obtained purely by chance. This is doubly true since, as we discussed earlier, randomness is characterised by an infinity of properties, so one must carefully choose the tests one performs.

The issues arising when testing RNGs can be illustrated pointedly with an example. Imagine a device which deterministically outputs the digits of the binary expansion of $\pi = \pi_1\pi_2\pi_3\ldots$ starting from the 10^{10}th bit. If we are unaware of the behaviour of this box and believe it to be a RNG, its output will appear extremely random to us; indeed, π passes all standard statistical tests of randomness [43] despite the fact that it is not known to even be Borel normal [59]. Nevertheless, the sequence produced by this box would be computable and thus not random at all.

Standard statistical tests of randomness focus on properties of the distribution of bits or bit strings within sequences, properties more closely related to Borel normality than algorithmic notions of randomness. Many such tests were developed with the aim of testing PRNGs, where reproducing such statistical predictions is a primary issue, particularly since failing to do so may leak information about the seed and thus break the security of the PRNG [39]. QRNGs have generally been tested against similar tests, such as the NIST [49] and DIEHARD [41] batteries, and generally perform well. For example, Quantis is officially certified as passing these tests on 1000 samples of 1 million bits [35]. Such tests, however, are far from confirming the randomness of the device; indeed, analysis of longer sequences (of 2^{32} bits) revealed (albeit it very small) bias and correlation amongst the output bits [4].

Such statistical non-uniformity is, however, to be expected in RNGs exploiting physical phenomena due to experimental imperfections and instability [7]. Inasmuch as this form of non-uniformity is small enough for the required application, this is not necessarily problematic as long as a QRNG remains certified by value indefiniteness: unlike for PRNGs, where non-uniformity is often a symptom of deeper issues, for QRNGs the unpredictability is guaranteed by the indeterministic nature of the device. Moreover, bias can be reduced by post-processing [46, 58], allowing quantum indeterminism to still be exploited sufficiently. Although testing such properties is crucial in order to ensure any bias remains tolerably low, such tests do not directly probe crucial advantages of quantum randomness, such as a degree of algorithmic randomness of their output.

While it is not possible to directly compute the algorithmic information content of strings produced by RNGs, one may still ask whether there are tests that indirectly probe this to try and differentiate PRNGS – which always produce computable sequences – from QRNGs. In the following sections we investigate more closely this question.

6 Experimental evidence of incomputability

In this section we describe several tests based on algorithmic properties which we use to investigate the properties of random bits obtained from both PRNGs and the QRNG detailed in Fig. 2. We tested 10 sequences of 2^{29} bits each obtain from sequences obtained from the initial bits of π – which can be seen as a form of pseudo-randomness [13] – as well as the PRNGs used by Python (Mersenne Twister algorithm) [1], Random123 [50], PCG [45], xoroshilro128+ [42], as well as the QRNG described in Sec. 4.1.

6.1 Tests of Borel normality

The notion of Borel normality was the first mathematical definition of algorithmic randomness [18], and although, like many standard tests of randomness, it focuses on the distribution of bits within a sequence, it is nonetheless worth looking at in its own right.

An infinite sequence $\mathbf{x} \in \{0,1\}^\infty$ is (Borel) normal if every binary string appears in the sequence with the right frequency (which is 2^{-n} for a string of length n). Every algorithmic random infinite sequence is Borel normal [22], but the converse implication is not true: there exist computable normal sequences, such as Champernowne's sequence mentioned earlier. Normality is invariant under finite variations: adding, removing, or changing a finite number of bits in any normal sequence leaves it normal.

The notion of normality was subsequently transposed from infinite sequences to (finite) strings [22]. In doing so, one has to replace limits with inequalities, and one obtains the following definition. For any fixed integer $m > 1$, consider the alphabet $B_m = \{0,1\}^m$ consisting of all binary strings of length m, and for every $1 \leq i \leq 2^m$ denote by N_i^m the number of occurrences of the lexicographical ith binary string of length m in the string x (considered over the alphabet B_m). By $|x|_m$ we denote the length of x over B_m; $|x|_1 = |x|$. A string $x \in B_m$ is *Borel normal (with accuracy* $\frac{1}{\log_2}$*)* if for every natural $1 \leq m \leq \log_2 \log_2 |x|$, we have:

$$\left| \frac{N_j^m(x)}{|x|_m} - 2^{-m} \right| \leq \frac{1}{\log_2 |x|}, \tag{2}$$

for every $1 \leq j \leq 2^m$. Almost all algorithmic random strings are Borel normal with accuracy $\frac{1}{\log_2}$, [22]; in particular, they have approximately the same numbers number of 0s and 1s. Furthermore, if all prefixes of a sequence are Borel normal, then the sequence is also Borel normal. The need to state the accuracy with which a string is Borel normal for finite strings arises from the fact that the right-hand-side of Eq. (2) may be replaced by any computable real function in $|x|$ converging to 0 for the definition to behave properly. The choice of accuracy thus allows one to consider normality quantitatively; we choose here the accuracy $\frac{1}{\log_2}$ by convention, and this will make little difference since we are interested in the relative normality of different sequences.

In Figure 3, $\left| \frac{N_j^m(x)}{|x|_m} - 2^{-m} \right| \log_2 |x|$ has been calculated, where $m = 1, 2, \ldots, 5$. A value exceeding 1 indicates that a sequences fails the normality test at accuracy $\frac{1}{\log_2}$. Although we see that all sequences pass the test at this accuracy, it is evident that the sequence produced by the QRNG is significantly less normal than the pseudorandom sequences. This is, however, unexpected, since the experiment implementing the QRNG was known to exhibit bias due to experimental imperfections [38] and, as discussed in the previous section, such a bias is not necessarily problematic for a QRNG as long as it is sufficiently small for the relevant application.

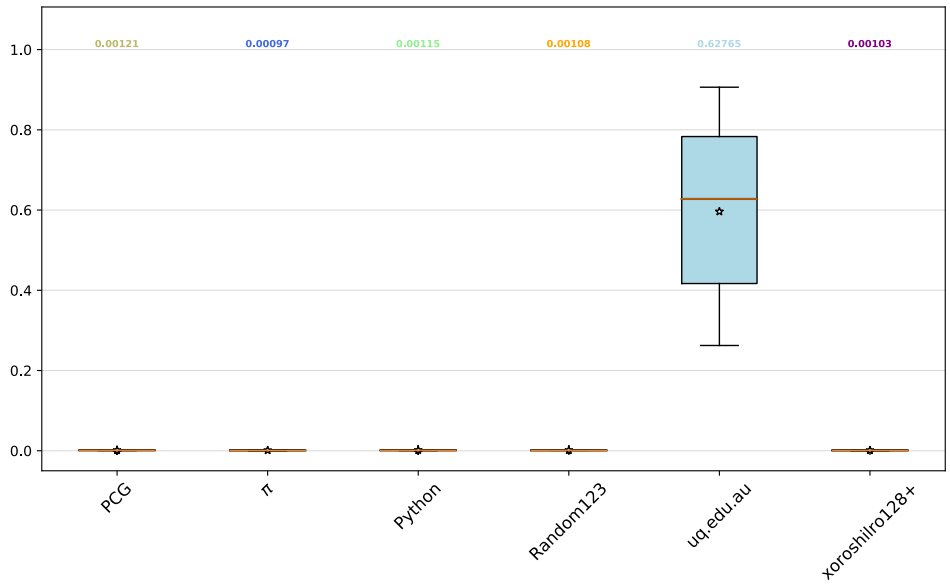

Figure 3: Borel normality test: Box-plot showing the distribution of the quantity $\left|\frac{N_j^m(x)}{|x|_m} - 2^{-m}\right| \log_2 |x|$ for the strings produced by the various RNGs. The averages for each source are marked at the top of the plot.

6.2 A test of incomputability

Is it possible to possible to give a test which rejects the randomness for every computable sequence? The answer is affirmative: such a Martin-Löf test exists (for more technical details on Martin-Löf tests and algorithmic randomness, see [22]). To specify the test we must define the sequences of its nth component for all integers $n > 0$. The nth component is the union of all $\{0,1\}^*\{0,1\}^\infty$ for which there is an e such that $\sigma(0) = \varphi_e(0), \ldots, \sigma(e+n+1) = \varphi_e(e+n+1)$ and $\sigma \in \{0,1\}^*$. This is an open computably enumerable class which that contains all computable sets, as each computable set has a characteristic function φ_e. Furthermore, the measure of the nth component is bounded from above by the $\sum 2^{-n-e-2}$, which in turn is bounded from above by 2^{-n-1}, as the σ derived from φ_e has length $e+n+2$ and is a prefix of the set for which φ_e computes the characteristic function.

It is not difficult to see that the above test depends on the enumeration φ_e, and there is no obvious "natural" choice. Furthermore, invariance under finite variations renders the test unsuitable for finite experiments. As a result, it is necessary to consider more indirect methods to test the incomputability of sequences produced by RNGs.

6.3 The Chaitin-Schwartz-Solovay-Strassen test

In contrast with standard tests of randomness which check specific properties of randomness of strings of bits, the proposed test is based solely on the behaviour of random strings as selectors for the Solovay-Strassen probabilistic primality test [54].

To test whether a positive integer n is prime, we take k natural numbers uniformly distributed between 1 and $n-1$, inclusive, and, for each one i, check whether a certain, easy to compute, predicate $W(i,n)$ holds. If $W(i,n)$ is true then "i is a witness of n's compositeness", hence n is composite. If $W(i,n)$ holds for at least one i then n is composite; otherwise, the test is inconclusive, but in this case the probability that n is prime is greater than $1-2^{-k}$. This is due

to the fact that at least half the i's from 1 to $n-1$ satisfy $W(i,n)$ if n is indeed composite, and *none* of them satisfy $W(i,n)$ if n is prime [54].

Selecting k natural numbers between 2 and $n-1$ is equivalent to choosing a binary string s of length $n-2$ with k 1's such that the ith bit is 1 if and only if i is selected. Chaitin and Schwartz [26] proved that, if s is a long enough algorithmically random binary string, then n is prime iff $Z(s,n)$ is true, where Z is a predicate constructed directly from conjunctions of negations of W. The crucial fact is that the set of algorithmically random strings is highly incomputable: technically the set is immune, that is, it contains no infinite computably enumerable subset [22]. As a consequence, de-randomisation is non-constructive and not effective.

For our proposed tests, we focus on whether certain numbers succeed in witnessing the compositeness of Carmichael numbers. A Carmichael number is a composite positive integer k satisfying the congruence $b^{k-1} \equiv 1 \pmod{k}$ for all integers b relative prime to k. Carmichael numbers are composite, but are difficult to factorise and thus are "very similar" to primes; they are sometimes called pseudo-primes. Carmichael numbers can fool Fermat's primality test, but less the Solovay-Strassen test. Increasingly Carmichael numbers become "rare" [2].

The Chaitin-Schwartz-Solovay-Strassen test examines the computational capacity of a random string s when used by Solovay-Strassen primality test for Carmichael (composite) numbers. For our analysis, we have used all Carmichael numbers less than 2^{29} as computed in [47]. As a metric to compare the performance of strings we look at the number of wrong answers – that is, when the test determines incorrectly that a Carmichael number is prime – returned by the Solovay-Strassen probabilistic algorithm.

The Solovay-Strassen test is capable of giving empirical evidence of incomputability, in stark contrast to most tests of randomness. Indeed, the Borel normality test discussed before is unable to do so: the normality of Champernowne's sequence mentioned earlier is evidence of this.

In Figure 4 we plot the performance of various strings (the same ones as tested for Borel normality in Fig. 3) using the metric described above. The box-plot shows, for each source of bits, the average score over the 10 strings tested as well as the quartiles showing the spread of scores. We have seen that quantum random bits perform significantly better than the bits generated by all the PRNGs: they provide significantly fewer false claims of primality for the Carmichael numbers on average. As all the Carmichael numbers tests were eventually proved composite by all random strings, the quantum random bits are more "faster" in determining the compositeness of pseudo-primes because the number of tests are smaller.

6.4 Solovay-Strassen practicality randomness test

As a final test, we performed another test inspired by the Solovay-Strassen test that was previously used in [20] to try to distinguish quantum randomness from pseudo randomness. Here we again tested each Carmichael number n less than 2^{29}, as computed in [47], with witness numbers selected from the binary representation of $\lceil \lg n \rceil$ bits of the source string, skipping the numbers that were out-of-range. For this, the metric is given by the smallest k *[Why smallest, not average?]* such at most k witness numbers were required to obtain a verdict of non-primality for all of the 483 Carmichael numbers less than 2^{29}. This test determines, in practice, whether the random sequence of bits is more useful for deciding non-primality of a set of composite numbers with the metric being the total number of bits required (smaller the better). As each Carmichael number is tested we never recycled bits from the sample string; that is, we only reset the source string to the first bit when there was a need to try a larger value of k to pass this metric.

[2] There are 1,401,644 Carmichael numbers in the interval $[1, 10^{18}]$.

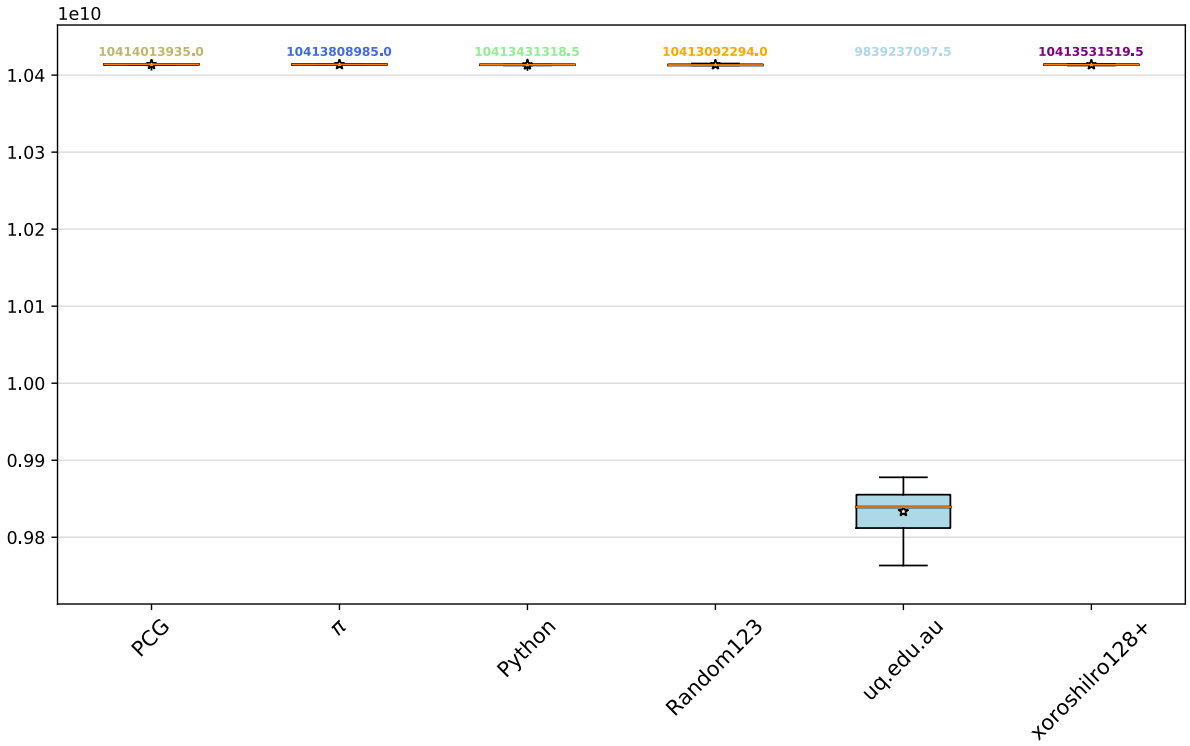

Figure 4: Chaitin-Schwartz-Solovay-Strassen test: Box-plot showing the distribution in the number of Carmichael numbers incorrectly identified as prime by the random sequences produced by the various RNGs. The averages for each source are marked at the top of the plot.

In Figure 5 we plot the distribution of this metric. We see that there is little different between the performance of the various source, except for the bits produced by the QRNG which again show an advantage over all the PRNGs. Compared to the Chaitin-Schwartz-Solovay-Strassen test of the previous section, however, this advantage is much less pronounced and only slightly fewer bits (based on the smaller required k) were, on average, needed by the bits produced by the QRNG compared to those produced by the PRNGs.

7 Conclusions

Acknowledgment

The authors acknowledge fruitful discussions with Arkady Fedorov, Anatolyt Kulikov, Frank Stephan and Karl Svozil.

References

[1] Generate pseudo-random numbers. https://docs.python.org/3/library/random.html, November 2017.

[2] Pseudorandom number generator. https://en.wikipedia.org/wiki/Pseudorandom_number_generator, 26 Novenber 2017.

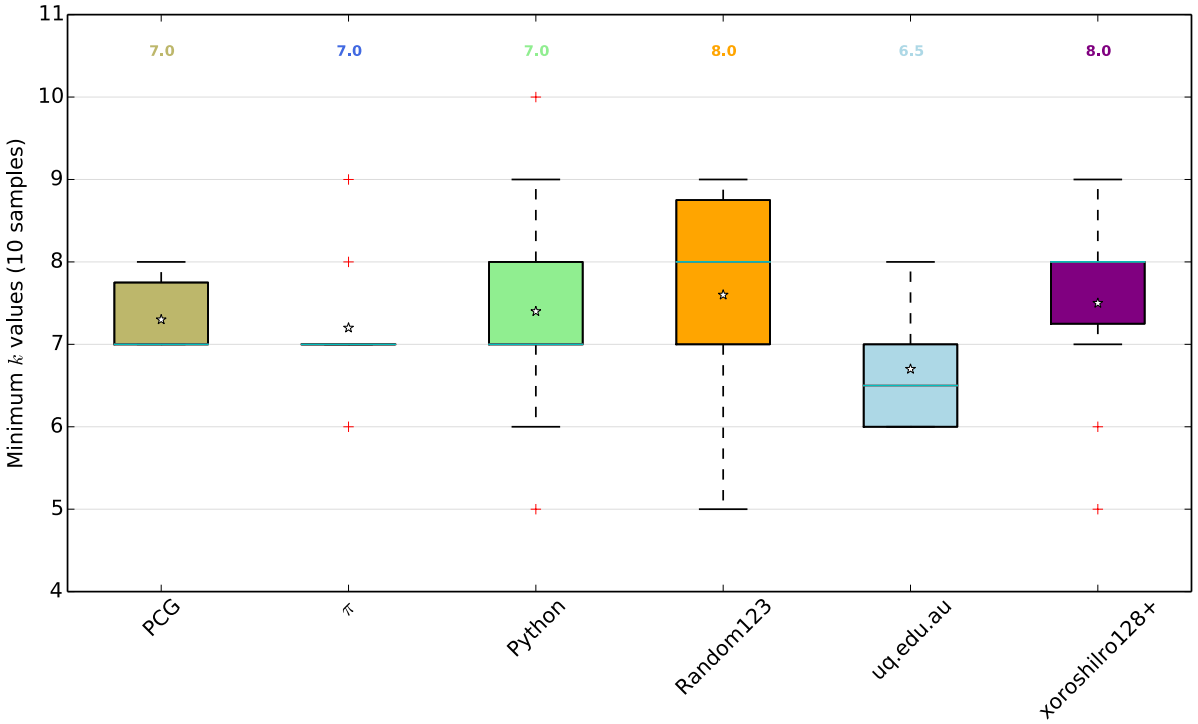

Figure 5: Solovay-Strassen test: Box-plot showing the distribution in the minimum number k of witnesses needed to verify the compositeness of all Carmichael numbers less than 2^{29}. The averages for each source are marked at the top of the plot.

[3] Alastair A. Abbott. *Value Indefiniteness, Randomness and Unpredictability in Quantum Foundations*. PhD thesis, University of Auckland; École Normale Supérieure de Paris, 2015.

[4] Alastair A. Abbott, Laurent Bienvenu, and Gabriel Senno. Non-uniformity in the Quantis random number generator. *CDMTCS Research Report Series 472*, 2014.

[5] Alastair A. Abbott, Cristian S. Calude, Jonathan Conder, and Karl Svozil. Strong Kochen-Specker theorem and incomputability of quantum randomness. *Physical Review A*, 86:062109, 2012.

[6] Alastair A. Abbott, Cristian S. Calude, and Karl Svozil. Value-indefinite observables are almost everywhere. *Physical Review A*, 89:032109, 2013.

[7] Alastair A. Abbott, Cristian S. Calude, and Karl Svozil. A quantum random number generator certified by value indefiniteness. *Mathematical Structures in Computer Science*, 24(3):e240303, 2014.

[8] Alastair A. Abbott, Cristian S. Calude, and Karl Svozil. A non-probabilistic model of relativised predictability in physics. *Information*, 6(4):773–789, 2015.

[9] Alastair A. Abbott, Cristian S. Calude, and Karl Svozil. On the unpredictability of individual quantum measurement outcomes. In Lev D. Beklemishev, Andreas Blass, Nachum Dershowitz, Bernd Finkbeiner, and Wolfram Schulte, editors, *Fields of Logic and Computation II – Essays Dedicated to Yuri Gurevich on the Occasion of His 75th Birthday*, volume 9300 of *Lecture Notes in Computer Science*, pages 69–86. Springer International, Switzerland, 2015.

[10] Alastair A. Abbott, Cristian S. Calude, and Karl Svozil. A variant of the Kochen-Specker theorem localising value indefiniteness. *Journal of Mathematical Physics*, 56:102201, 2015.

[11] Antonio Acín. True quantum randomness. In Antoine Suarez and Peter Adams, editors, *Is Science Compatible with Free Will?: Exploring Free Will and Consciousness in the Light of Quantum Physics and Neuroscience*, chapter 2, pages 7–22. Springer-Verlag, New York, 2013.

[12] Alain Aspect, Philippe Grangier, and Gérard Roger. Experimental realization of Einstein-Podolsky-Rosen-Bohm *Gedankenexperiment*: A new violation of Bell's inequalities. *Physical Review Letters*, 49(2):91–94, 1982.

[13] D. H. Bailey, J. M. Borwein, C. S. Calude, M. J. Dinneen, M. Dumitrescu, and A. Lee. An empirical approach to the normality of π. *Experimental Mathematics*, 21(4):375–384, 2012.

[14] Maya Bar-Hillel and Willem A Wagenaar. The perception of randomness. *Advances in Applied Mathematics*, 12(4):428–454, 1991.

[15] J. S. Bell. On the Eistein-Podolsky-Rosen paradox. *Physics*, 1(3):195–200, 1964.

[16] Manabendra Nath Bera, Antonio Acín, Marek Kuś, Morgan Mitchell, and Maciej Lewenstein. Randomness in quantum mechanics: Philosophy, physics and technology. *Reports on Progress in Physics*, 80:124001, 2017.

[17] Daniel J. Bernstein, Yun-An Chang, Chen-Mou Cheng, Li-Ping Chou, Nadia Heninger, Tanja Lange, and Nicko van Someren. Factoring RSA keys from certified smart cards: Coppersmith in the wild. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, pages 341–360, Berling, 2013. Springer.

[18] Émile Borel. Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo*, 27:247–271, 1909.

[19] C. S. Calude. Borel normality and algorithmic randomness. In G. Rozenberg and A. Salomaa, editors, *Developments in Language Theory*, pages 113–129. World Scientific, Singapore, 1994.

[20] C. S. Calude, M. J. Dinneen, M. Dumitrescu, and K. Svozil. Experimental evidence of quantum randomness incomputability. *Physical Review A*, 82:022102, 2010.

[21] Cristian Calude. *Information and Randomness—An Algorithmic Perspective*. Springer, Berlin, second edition, 2002.

[22] Cristian S. Calude. *Information and Randomness: An Algorithmic Perspective*. Springer-Verlag, Berlin, second edition, 2002.

[23] Cristian S. Calude. Quantum randomness: From practice to theory and back. In S. B. Cooper and M. Soskova, editors, *The Incomputable: Journeys Beyond the Turing Barrier*, pages 169–181. Springer, 2017.

[24] Cristian S. Calude and Karl Svozil. Quantum randomness and value indefiniteness. *Advanced Science Letters*, 1(2):165–168, 2008.

[25] G. J. Chaitin. Algorithmic information theory. *IBM Journal of Research and Development*, 21(4):350–359, 1977.

[26] G. J. Chaitin and J. T. Schwartz. A note on Monte Carlo primality tests and algorithmic information theory. *Communications on Pure and Applied Mathematics*, 31(4):521–527, 1978.

[27] D. G. Champernowne. The construction of decimals normal in the scale of ten. *Journal of the London Mathematical Society*, 8:254–260, 1933.

[28] Roger Colbeck and Adrian Kent. Private randomness expansion with untrusted devices. *Journal of Physics A: Mathematical and General*, 44:095305, 2011.

[29] Antony Eagle. Randomness is unpredictability. *The British Journal for the Philosophy of Science*, 56(4):749–790, 2005.

[30] Antony Eagle. Chance versus randomness. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Stanford University, Spring 2014 edition, 2014.

[31] Małgorzata Figurska, Maciej Stańczyk, and Kamil Kulesza. Humans cannot consciously generate random numbers sequences: Polemic study. *Medical Hypotheses*, 70(1):182–185, 2008.

[32] Oded Goldreich. *Foundations of cryptography I: Basic Tools*. Cambridge University Press, Cambridge, 2001.

[33] Ronald Graham and Joel H. Spencer. Ramsey theory. *Scientific American*, 262:112–117, September 1990.

[34] Alan Hájek. Interpretations of probability. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Stanford University, Winter 2012 edition, 2014.

[35] ID Quantique. Quantis QRNG. https://www.idquantique.com/random-number-generation/.

[36] T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, and A. Zeilinger. A fast and compact quantum random number generator. *Review of Scientific Instruments*, 71:1675–1680, 2000.

[37] Simon B. Kochen and Ernst Specker. The problem of hidden variables in quantum mechanics. *Journal of Mathematics and Mechanics (now Indiana University Mathematics Journal)*, 17(1):59–87, 1967.

[38] Anatoly Kulikov, Markus Jerger, Anton Potočnik, Andreas Wallraff, and Arkady Fedorov. Realization of a quantum random generator certified with the kochen-specker theorem. 2017.

[39] Arjen K. Lenstra1, James P. Hughes, Maxime Augier, Joppe W. Bos1, Thorsten Kleinjung, and Christophe Wachter. Ron was wrong, whit is right. Santa Barbara: IACR: 17, https://eprint.iacr.org/2012/064.pdf, 2012.

[40] Giuseppe Longo and Thierry Paul. The mathematics of computing between logic and physics. In S. B. Cooper and A. Sorbi, editors, *Computability in Context: Computation and Logic in the Real World*, chapter 7, pages 243–274. Imperial College Press/World Scientific, London, 2008.

[41] G. Marsaglia and A. Zaman. Towards a universal random number generator. *Statistics & Probability Letters*, 9(1):35–39, 1990. http://www.stat.fsu.edu/pub/diehard/.

[42] George Marsaglia. Xorshift rngs. *Journal of Statistical Software*, 8(14), 2003.

[43] George Marsaglia. On the randomness of Pi and other decimal expansions. *Interstat*, 10(5):1–17, October 2005.

[44] P. Martin-Löf. The definition of random sequences. *Information and Control*, 9(6):602–619, 1966.

[45] Melissa E. O'Neill. Pcg: A family of simple fast space-efficient statistically good algorithms for random number generation. Technical Report HMC-CS-2014-0905, Harvey Mudd College, Claremont, CA, Sep 2014.

[46] Y. Peres. Iterating von Neumann's procedure for extracting random bits. *The Annals of Statistics*, 20(1):590–597, 1992.

[47] R. G. E. Pinch. The Carmichael numbers up to 10^{21}. In Anne-Maria Ernvall-Hytönen, Matti Jutila, Juhani, Karhumäki, and Arto Lepistö, editors, *Proceedings of Conference on Algorithmic Number Theory 2007*, volume 46, pages 129–131, Finland, 2007. TUCS.

[48] S. Pironio, A. Acín, S. Massar, A. Boyer de la Giroday, D. N. Matsukevich, P. Maunz, S. Olmchenk, D. Hayes, L. Luo, T. A. Manning, and C. Monroe. Random numbers certified by Bell's Theorem. *Nature*, 464(09008), 2010.

[49] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Special Publication 800-22, NIST, 2010.

[50] John K. Salmon, Mark A. Moraes, Ron O. Dror, and David E. Shaw. Parallel random numbers: As easy as 1, 2, 3. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC11)*, New York, 2011. ACM.

[51] H. Schmidt. Quantum-mechanical random-number generator. *Journal of Applied Physics*, 41(2):462–468, 1970.

[52] Y. Shen, L. Tian, and H. Zou. Practical quantum random nubmer generator based on measuring the shot noise of vacuum states. *Physical Review A*, 81(063814), 2010.

[53] R. Solovay and V. Strassen. Erratum: A fast Monte-Carlo test for primality. *SIAM Journal on Computing*, 7(1):118, 1977.

[54] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM Journal on Computing*, 6(1):84–85, 1977. Corrigendum in Ref. [53].

[55] A. Stefanov, N. Gisin, O. Guinnard, L. Guinnard, and H. Zbinden. Optical quantum random number generator. *Journal of Modern Optics*, 47(4):595–598, 2000.

[56] M. Stipčević and B. M. Rogina. Quantum random number generator based on photonic emission in semiconductors. *Review of Scientific Instruments*, 78(4):045104, 2007.

[57] K. Svozil. The quantum coin toss — testing microphysical undecidability. *Physics Letters A*, 143(9):433–437, 1990.

[58] J. von Neumann. Various techniques used in connection with random digits. *National Bureau of Standards Applied Math Series*, **12** (1951), 36–38. In A. H. Traub, editor, *John von Neumann, Collected Works*, pages 768–770. MacMillan, New York, 1963.

[59] Stan Wagon. Is π normal? In J. Lennart Berggren, Jonathan M. Borwein, and Peter B. Borwein, editors, *Pi: A Source Book*, pages 557–559. Springer-Verlag, New York, 2004.

Appendix: Data and numerical results