COMPSCI750: Computational Complexity

Cristian S. Calude

Semester 2, 2022

Understanding and generating randomness

Randomness

Random numbers are used extensively in various applications in science, economy and security, ranging from cryptography to numerical simulations.

'Good' randomness is the critical resource for all these applications, but high quality randomness is neither easy to produce nor to certify.

Kleroterion

was a randomisation device used by the Athenian city during the period of democracy to select citizens to most state offices and court juries.

Kleroterion by Marsyas – Own work, CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=475523

Kleroterion

Bronze identification tickets were inserted to indicate eligible jurors who were also divided into tribes. By a random process, a whole row would be accepted or rejected for jury service.

In front of each court there was a Kleroterion.

This method, called *sortition*, has been used in Belgium parliaments since 2017.

Randomness appears and is necessary almost everywhere

Random numbers had been around for more than 4,000 years, but never have they been in such demand as in our time. The oldest known dice were discovered in a 24th century B.C. tomb in the Middle East.

In 1890 statistician Francis Galton wrote that

> *As an instrument for selecting at random, I have found nothing superior to dice.*

However, the modern world demanded a lot more randomness than dice could offer.

John von Neumann developed a *pseudo-random generator* (PRNG) around 1946: start with an initial random seed value, square it, and slice out the middle digits. A sequence obtained by repeatedly using this method exhibits *some* statistical properties of randomness.

Randomness appears and is necessary almost everywhere

RAND Corporation machine generated numbers using a random pulse generator: its book *A Million Random Digits with 100,000 Normal Deviates* was published in 1955.

In 1951, Ferranti Mark 1 was the first computer with a built-in random number instruction—designed by A. Turing—that could generate 20 random bits at a time using electrical noise.

All modern computers have algorithms that generate *pseudo-random numbers*.

What is random?

> *I am convinced that the vast majority of my readers, and in fact the vast majority of scientists and even nonscientists, are convinced that they know what 'random' is. A toss of a coin is random; so is a mutation, and so is the emission of an alpha particle. . . . Simple, isn't it? said Kac in [10].*

Well, no! Kac knew very well that randomness could be called many things, but not simple, and in fact his essay shows that randomness is complicated, and it can be described in more than one way

Books on probability theory do not even attempt to define it. *It's like the concept of a point in geometry books.*

What's wrong with these random numbers?

Pseudo-random numbers are used hundreds of billions of times a day to encrypt data in electronic networks.

The pitfalls of low quality randomness have been discovered in the Internet era. An example is the discovery in 2012 of a weakness in the encryption system used worldwide for online shopping, banking and email; the flaw was traced to the numbers a PRNG has produced [12].

Do "better" random numbers exist?
Yes.

Can other methods produce "better" random numbers?
Yes.

Does true or perfect random number exist?
No.

Symptoms of random

Randomness is best understood through various "symptoms". Here are three of the largely accepted ones:

(i) *Unpredictability:* The impossibility of any agent, acting via uniform, effective means, to predict correctly and reproducibly the outcome of an experiment using finite information extracted from the environment.

(ii) *Incompressibility:* The impossibility to compress a random sequence.

(iii) *Typicality:* Random sequences pass every statistical test of randomness.

The paradox of randomness

The French mathematician Émile Borel, a pioneer of probability theory, argued that there is no way to formalise in an acceptable way the concept of randomness. His argument is based only one symptom of randomness, typicality, and is known as the *randomness paradox*.

A random bit-string should be “typical”: it should not stand out from the crowd of other bit-strings.

Here is Borel’s argument. Assume that there is a precise way to distinguish between “random bit-strings” and bit-strings which are “non-random”. It does not matter how this criterion was found or operates.

The paradox of randomness

Assume we have a precise **criterion** which can be applied to any bit-string and once a bit-string is given, we can say whether the bit-string is *random* or *non-random*.

Can the **criterion** be consistent? The answer is **negative**.

Indeed, choose the first bit-string which **criterion** asserts it is random. This particular bit-string is

> *the first bit-string satisfying the property of being random,*

a property making it atypical, so non-random!

communications to Monte-Carlo simulation.

There are many random number generators:

1. pseudo-random generators, software produced
   - PCG, Random123, xoroshilro128+
2. hardware generators, devices that generate random numbers from physical processes
   - macroscopic, e.g. coin, dice, roulette wheels, lottery machines,
   - microscopic, e.g. thermal noise, photoelectric effect, quantum effects.

In particular, there are many quantum random generators, from lab experiments to openly accessible on the internet and commercial like Quantis.

Quantum randomness

According to NIST, New quantum method generates really random numbers, phys.org/news/, [13, 6]:

“Quantum mechanics provides a superior source of randomness because measurements of some quantum particles (those in a ”superposition” of both 0 and 1 at the same time) have fundamentally unpredictable results.”

But,

- what does it mean superior?
- is the answer “because measurements of some quantum particles have fundamentally unpredictable results” good enough?

Why do we need another quantum random generator?

Because no current quantum random generator (QRNG) is provably better than the other generators, in particular, pseudo-random generators.

Is this so? Many advantages promised by QRNGs rely on the belief that the outcomes of quantum measurements are

intrinsically/irreducibly unpredictable.

corroborated but not proved by evidence.

This belief underlies:

- the use of QRNGs to produce “quantum random” sequences that are “truly unpredictable”,
- the generation of cryptographic keys unpredictable to any adversary.

Is this belief reasonable?

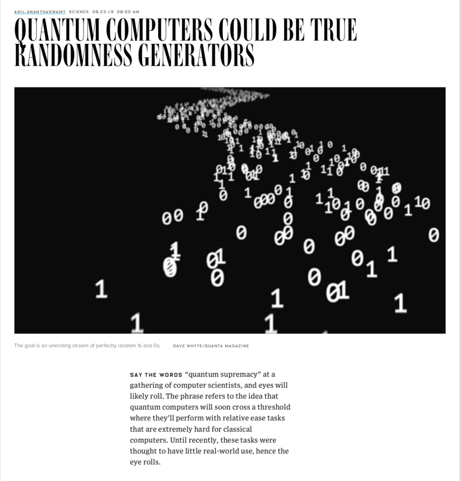
True randomness

Work reported in *Nature* (doi:10.1038/news.2010.181, 14 April 2010) claims that quantum randomness is true randomness:

Truly random numbers have been generated at last.

True randomness

More recently, an article published in July 2019 in Wired magazine claims that quantum computers could produce true randomness:



ANIL ANANTHASWAMY SCIENCE 06.23.19 08:00 AM

QUANTUM COMPUTERS COULD BE TRUE RANDOMNESS GENERATORS

The goal is an unending stream of perfectly random 1s and 0s. DAVE WHYTE/QUANTA MAGAZINE

SAY THE WORDS "quantum supremacy" at a gathering of computer scientists, and eyes will likely roll. The phrase refers to the idea that quantum computers will soon cross a threshold where they'll perform with relative ease tasks that are extremely hard for classical computers. Until recently, these tasks were thought to have little real-world use, hence the eye rolls.

True randomness

If “true randomness” means “lack of any correlations” or “maximal randomness”, does it exist?

What is the answer?

Proof:?

Questions

- How can we produce quantum randomness?
- What is the physical “reason” of quantum randomness?
- What is the quality of quantum randomness?
- Is quantum randomness better than pseudo-randomness?

Indeterminism and quantum randomness

- Quantum randomness is
  - postulated and
  - generally reduced to the indeterminism of quantum measurements: because the outcome is indeterministic there is no way to predict it, hence it is random
- However, indeterminism does not imply randomness and randomness does not imply indeterminism:
  - pseudo-randomness
  - coin-tossing (chaoticity)
  - Schrödinger equation

Is quantum randomness provable better than pseudo-randomness?

How should we formulate mathematically the question in the title?

What is the answer?

Proof:?

Value indefiniteness is a central notion to this discussion. Informally, for a given quantum system in a particular state, we say that an observable is value definite if the measurement of that observable is predetermined to take a (potentially hidden) value.

In defining it we are guided by Einstein, Podolsky and Rosen famous analysis in [9] which can be encapsulated in the following

> *EPR principle: If, without in any way disturbing a system, we can predict with certainty the value of a physical quantity, then there exists a definite value prior to observation corresponding to this physical quantity.*

See more in [1, 3].

Value indefiniteness

EPR principle justifies also

> *Eigenstate principle: If a quantum system is prepared in a state $|\psi\rangle$, then the projection observable P_ψ is value definite.*

Main assumptions

- Admissibility: Definite values must not contradict the statistical quantum predictions for compatible observables on a single quantum.
- Noncontextuality of definite values: The outcome obtained by measuring a value definite observable (a preexisting physical property) is *noncontextual*, i.e. it does not depend on other compatible observables which may be measured alongside the value definite observable.
- Eigenstate principle: If a quantum system is prepared in the state $|\psi\rangle$, then the projection observable P_ψ is value definite.

Assume the above three hypotheses.

Theorem [1]

Assume a quantum system prepared in the state $|\psi\rangle$ in dimension $n \geq 3$ Hilbert space $\mathfrak{C}^n$, and let $|\phi\rangle$ be any state neither orthogonal nor parallel to $|\psi\rangle$. Then the projection observable P_ϕ is value indefinite.

Value indefiniteness and unpredictability

> *EPR principle: If, without in any way disturbing a system, we can predict with certainty the value of a physical quantity, then there exists a definite value prior to observation corresponding to this physical quantity.*

Conversely, if no unique element of physical reality corresponding to a particular physical quantity exists, this is reflected by the physical quantity being value indefinite.

If a physical property is value indefinite we cannot predict with certainty the outcome of any experiment measuring this property.

Philosophical digression: Popper's unpredictability

If we assert of an observable event that it is unpredictable we do not mean, of course, that it is logically or physically impossible for anybody to give a correct description of the event in question before it has occurred; for it is clearly not impossible that somebody may hit upon such a description accidentally. What is asserted is that certain rational methods of prediction break down in certain cases—the methods of prediction which are practised in physical science.

Unpredictability revisited

We present a non-probabilistic model of prediction based on the ability of a computable operating agent to correctly predict using finite information extracted from the system of the specified experiment, [4].

Predictions should remain correct in any arbitrarily long (but finite) set of repetitions of the experiment.

A model of prediction

Consider a physical experiment E producing a single bit.
We consider an experiment E producing a single bit $x \in \{0, 1\}$.

An example of such an experiment is the measurement of a photon's polarisation after it has passed through a 50-50 beam splitter.

With a particular trial of E we associate the parameter λ (the state of the universe) which fully describes the trial. While λ is not in its entirety an obtainable quantity, we can view it as a resource that one can extract finite information from in order to predict the outcome of the experiment E.

A model of prediction (cont.)

An extractor is a (deterministic) function $\lambda \mapsto \langle\lambda\rangle$ mapping reals to rationals.

For example, $\langle\lambda\rangle$ may be an encoding of the result of the previous trial of E, or the time of day the experiment is performed.

A predictor for E is an algorithm (computable function) P_E which halts on every input and outputs **0** or **1** or **prediction withheld**.

P_E can utilise as input the information $\langle\lambda\rangle$, but, *as required by* EPR, must be passive, i.e. must not disturb or interact with E in any way.

A model of prediction (cont.)

A predictor P_E provides a correct prediction using the extractor $\langle\,\rangle$ for a trial of E with parameter λ if, when taking as input $\langle\lambda\rangle$, it outputs $\mathbf{0}$ or $\mathbf{1}$ and this output is equal to the result of the experiment.

The predictor P_E is $k, \langle\,\rangle$-correct if there exists an $n \geq k$ such that when E is repeated n times with associated parameters $\lambda_1, \ldots, \lambda_n$ producing the outputs $x_1, x_2, \ldots, x_n$, P_E outputs the sequence $P_E(\langle\lambda_1\rangle), P_E(\langle\lambda_2\rangle), \ldots, P_E(\langle\lambda_n\rangle)$ with the following two properties:

(i) no prediction in the sequence is incorrect, and

(ii) in the sequence there are k correct predictions.

A model of prediction (cont.)

The outcome x of a single trial of the experiment E performed with parameter λ is predictable (with certainty) if there exist an extractor $\langle\,\rangle$ and a predictor P_E which is

1. $k, \langle\,\rangle$-correct for all k, and
2. $P_E(\langle\lambda\rangle) = x$.

Unpredictability of individual measurements

Theorem 3

If E is an experiment measuring a quantum value indefinite observable, then every predictor P_E using any extractor $\langle\,\rangle$ is not $k, \langle\,\rangle$-correct, for all k.

Theorem 4

In an infinite repetition of E as considered above, no single bit x_i of the generating infinite sequence $x_1 x_2 \ldots$ can be predicted.

Incomputability of a form of quantum randomness

> **epr principle**: *If a repetition of measurements of an observable generates a computable sequence, then this implies these observables were value definite.*

Theorem 5 [1]

Assume the epr and Eigenstate principles. An infinite repetition of the experiment E measuring a quantum value indefinite observable generates an incomputable (even bi-immune, i.e. no algorithm can compute correctly more than finite many digits of the) infinite sequence $x_1 x_2 \ldots$.

Generalised beam QRNG producing maximum unpredictable, incomputable sequences

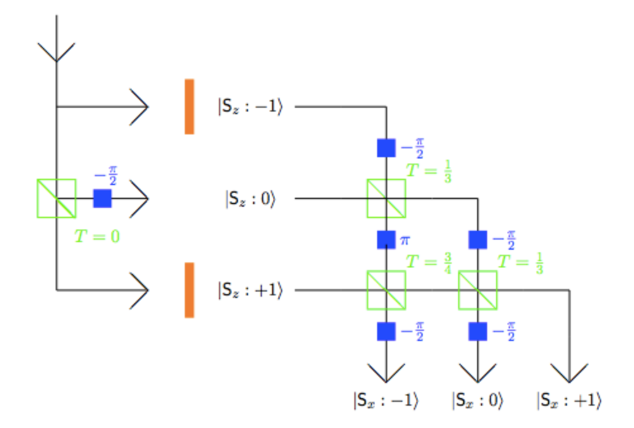


Figure: Generalised beam QRNG

Realisation of the generalised beam 3D QRNG with qutrits

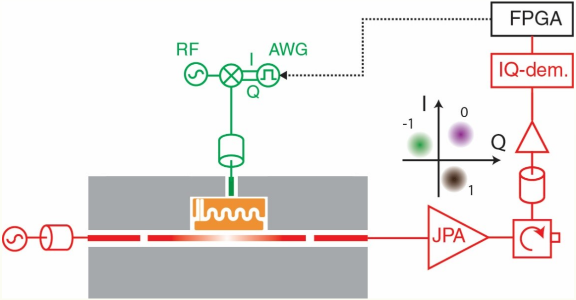


Figure: Realisation of a QRNG with superconducting circuits on a chip (gray). A qutrit (yellow) is coupled to a read out (red) and control circuitries. Near quantum limited Josephson parametric amplifier (JPA) can be used to discriminate all possible outcomes of the protocol with certainty including false runs. Fast programmable gate array (FPGA) can be used to provide the correction pulses to reinitialise the qutrit in its ground state right after the end of the protocol run. [11]

Experimental testing of the quality of the generalised beam QRNG with qutrits

The theory developed above guarantees the bi-immunity and unpredictability of every sequence produced by the generalised beam 3D QRNG with qutrits.

The experimental analysis of the quantum random bits [2] reported only weak evidence of incomputability. Some possible reasons are;

1. a problematic branch with probability zero used in the generalised beam splitter,
2. not long enough length of samples,
3. imperfections in the implementation of the measuring protocol.

An new 3D QRNG with a stronger theoretical certification which, hopefully, will be experimentally confirmed is in [5].

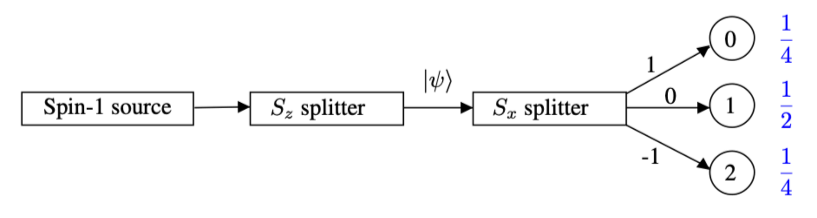


Figure: 3D QRNG with probabilities $1/4, 1/2, 1/4$.

Transforming 3D strings into 2D strings

As many applications require binary random strings, the following computable alphabetic morphism $\varphi : \{0,1,2\} \to \{0,1\}$

$$\varphi(a) = \begin{cases} 0, & \text{if } a = 0, \\ 1, & \text{if } a = 1, \\ 0, & \text{if } a = 2, \end{cases}$$

transforms by sequential concatenation ternary strings/sequences into binary ones and preserves the validity of Theorems 3,4, 5.

Testing incomputability: Solovay-Strassen test

The Solovay-Strassen test checks the primality of a positive integer n: take k natural numbers uniformly distributed between 1 and $n-1$, inclusive, and, for each $i(= i_1, \ldots, i_k)$, check whether a certain, easy to compute, the Solovay-Strassen predicate $W(i,n)$ holds:

1. If $W(i,n)$ is true then "i is a witness of n's compositeness", hence n is composite.
2. 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 between 1 and $n-1$ satisfy $W(i,n)$ if n is composite, and *none* of them satisfy $W(i,n)$ if n is prime [15].

Testing incomputability: Chaitin-Schwartz Theorem

Consider $s = s_0 \dots s_{m-1}$ a binary string (of length m) and n an integer greater than 2. Let k be the smallest integer such that $(n-1)^{k+1} > 2^m - 1$; we can thus rewrite the number whose binary representation is s into base $n-1$ and obtain the unique string $d_k d_{k-1} \dots d_0$ over the alphabet $\{0, 1, \dots, n-2\}$, that is,

$$\sum_{i=0}^{k} d_i (n-1)^i = \sum_{t=0}^{m-1} s_t 2^t.$$

The predicate $Z(s, n)$ is defined by

$$Z(s, J) = \neg W(1 + d_0, n) \wedge \cdots \wedge \neg W(1 + d_{k-1}, n), \qquad (1)$$

where W is the Solovay-Strassen predicate.

Testing incomputability: Chaitin-Schwartz Theorem

Theorem *For all sufficiently large c, if s is a c-incompressible string of length $\ell(\ell+2c)$ and n is an integer whose binary representation is ℓ bits long, then $Z(s,n)$ is true if and only if n is prime.*

The crucial fact is that the set of c-incompressible binary strings is highly incomputable: technically the set is immune, that is, it contains no infinite computably enumerable subset [7]. As a consequence, de-randomisation is thus *non-constructive*, and thus without practical value.

We will use Carmichael numbers as these target composites.

A Carmichael number is a composite positive integer n satisfying the congruence $b^{n-1} \equiv 1 \pmod{n}$ for all integers b relatively prime to n. Although Carmichael numbers are composite, they are difficult to factorise and thus are "very similar" to primes; they are sometimes called pseudo-primes.

Many Carmichael numbers can pass Fermat's primality test [16], but less of them pass the Solovay-Strassen test. Increasingly Carmichael numbers become "rare": there are 1,401,644 Carmichael numbers in the interval $[1, 10^{18}]$.

Testing incomputability with Chaitin-Schwartz Theorem

In order to carry out the test we first fix c. For each Carmichael number n (with an ℓ-bit binary representation) we take $c = \ell - 1$. This choice is somewhat arbitrary; other choices could of course be made but would make little difference to our test.

Testing incomputability with Chaitin-Schwartz Theorem

For each n we take $\ell\,(\ell + 2c)$ bits.
Rewriting s in base $n-1$ as above, we then compute $W(1+d_j, n)$ for $0 \leq j \leq k$ until the first j is found such that $W(1+d_j, n)$ holds (and the compositeness of n is thus witnessed).

The test looks for direct violations of the Chaitin-Schwartz Theorem: a violation appears when for all $j = 0, \ldots, k-1$, $W(1+d_j, n)$ are false; that is, all tests wrongly conclude that n is "probably prime".

Testing incomputability with Chaitin-Schwartz Theorem

As the Solovay-Strassen test guarantees that $W(1+d_j, n)$ is true with probability at least one-half when n is a composite number, it quickly becomes difficult, in practice, to observe such violations for even the smallest Carmichael numbers used in the previous tests.

In order to observe some violations we restrict the test to the odd composite numbers less than 50: $9, 15, 21, 25, 27, 33, 35, 39, 45, 49$. For these numbers, we compute $Z(s, n)$ by reading $\ell(\ell + 2c)$ bits and following the same procedure as in the third test. When $Z(s, n) = 1$, a violation of the Chaitin-Schwartz Theorem is thus observed.

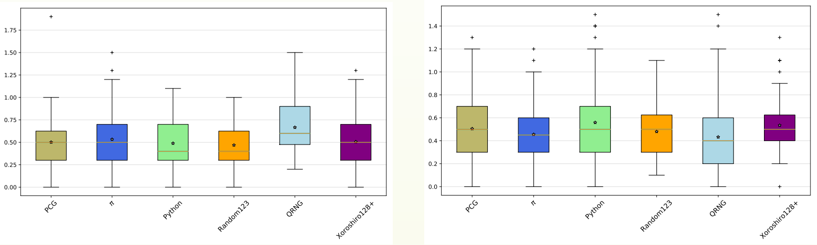
Testing incomputability with Chaitin-Schwartz Theorem

Since testing this predicate a single time on the ten numbers above would give insufficient statistics to observe any difference between the sources, we then repeated the above procedure reading from the 2nd bit of each string, then the 3rd, etc., until all the random bits have been used.

The metric is thereby taken as the average number of violations observed for the 10 composites tested (where the average is taken over all the repetitions), [2].

Testing incomputability with Chaitin-Schwartz Theorem

Figures (a) and (b) show the results of this test for the 80 strings of each of the six sources used in the previous tests: again, the tests in the former figure use the original strings from each source while the tests in the latter use the complemented strings.



(a) (b)

Figure: Distribution of the average count of violations of the Chaitin-Schwartz Theorem for all odd composite numbers less than 50 by (a) the 80 strings from each RNG, and (b) their complements.

Testing incomputability with Chaitin-Schwartz Theorem

The results of the Kolmogorov-Smirnov tests [8], to determine whether there are any statistically significant differences, for the data in the above Figures show that the QRNG exhibits significantly different behaviour on the original (a), but no significant difference in (b). The reason is unclear, and further investigation is required, [2].

Testing the assumptions

Are the main assumptions, Admissibility, Non-contextuality, Eigenstate and epr principles, used in the proof of Theorems 3,4,5, physically “acceptable”?

A cautiously affirmative answer to this question comes from the results of the testing incomputability of strings of length 2^{32} with Chaitin-Schwartz Theorem in [2].

References

[1] A. A. Abbott, C. S. Calude, J. Conder, and K. Svozil.
Strong Kochen-Specker theorem and incomputability of quantum randomness.
Physical Review A, 86(062109), Dec 2012.

[2] A. A. Abbott, C. S. Calude, M. J. Dinneen, and N. Huang.
Experimentally probing the algorithmic randomness and incomputability of quantum randomness.
Physica Scripta, 94(4):045103, Feb 2019.

[3] A. A. Abbott, C. S. Calude, and K. Svozil.
Value indefiniteness is almost everywhere.
Physical Review A, 89(3):032109–032116, 2014.

References (cont.)

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

[5] J. M. Agüero Trejo and C. S. Calude.
A new quantum random number generator certified by value indefiniteness.
Theoretical Computer Science, 862:3–13, Mar. 2021.

[6] P. Bierhorst, E. Knill, S. Glancy1, Y. Zhang, A. Mink, S. Jordan, A. Rommal, Y.-K. Liu, B. Christensen, S. W. Nam, M. J. Stevens, and L. K. Shalm.
Experimentally generated randomness certified by the impossibility of superluminal signals.
Nature, 556:223–226, April 2018.

References (cont.)

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

[8] W. J. Conover.
Practical Nonparametric Statistics.
John Wiley & Sons, New York, 1999.

[9] A. Einstein, B. Podolsky, and N. Rosen.
Can quantum-mechanical description of physical reality be considered complete?
Physical Review, 47(10):777–780, May 1935.

[10] M. Kac.
What is random?
American Scientist, 71:405–406, 1983.

References (cont.)

[11] A. Kulikov, M. Jerger, A. Potočnik, A. Wallraff, and A. Fedorov.
Realization of a quantum random generator certified with the Kochen-Specker theorem.
Phys. Rev. Lett., 119:240501, Dec 2017.

[12] A. K. Lenstra, J. P. Hughes, M. Augier, J. W. Bos, T. Kleinjung, and C. Wachter.
Ron was wrong, Whit is right.
Santa Barbara: IACR: 17, https://eprint.iacr.org/2012/064.pdf, 2012.

[13] S. Pironio.
The certainty of randomness.
Nature, 556:176–177, 2018.

References (cont.)

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

[15] R. Solovay and V. Strassen.
A fast Monte-Carlo test for primality.
SIAM Journal on Computing, 6(1):84–85, 1977.
Corrigendum in Ref. [14].

[16] A. A. Tokuç.
Fermat Probabilistic Test, May 2021.